Probabilistic Models of Visual Appearance for Object Identity, Class, and Pose Inference

A dissertation presented by **Damien Teney**

Directed by **Prof. Justus Piater**

and **Prof. Jacques G. Verly**

Submitted to the University of Liège in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science.

Jury	Prof. A. Leonardis, University of Birmingham
	Prof. J. Little, University of British Columbia
	Prof. J. Piater, University of Innsbruck
	Prof. J. Verly, University of Liège
	Prof. L. Wehenkel, University of Liège

Faculté des Sciences Appliquées
Département d'Electricité, Electronique et Informatique

Abstract

The topic of object recognition is a central challenge of computer vision. In addition to being studied as a scientific problem in its own right, it also counts many direct practical applications. We specifically consider robotic applications involving the manipulation, and grasping of everyday objects, in the typical situations that would be encountered by personal service robots. Visual object recognition, in the large sense, is then paramount to provide a robot the sensing capabilities for scene understanding, the localization of objects of interests and the planning of actions such as the grasping of such objects.

This thesis presents a number of methods that tackle the related tasks of object detection, localization, recognition, and pose estimation in 2D images, of both specific objects and of object categories. We aim at providing techniques that are the most generally applicable, by considering those different tasks as different sides of a same problem, and by not focusing on a specific type of image information or image features. We first address the use of 3D models of objects for continuous pose estimation. We represent an object by a constellation of points, corresponding to potentially observable features, which serve to define a continuous probability distribution of such features in 3D. This distribution can be projected onto the image plane, and the task of pose estimation is then to maximize its "match" with the test image. Applied to the use of edge segments as observable features, the method is capable of localizing and estimating the pose of non-textured objects, while the probabilistic formulation offers an elegant way of dealing with uncertainty in the definition of the models, which can be learned from observations — as opposed to being available as hand-made CAD models. We also propose a method, framed in a similar probabilistic formulation, in order to obtain, or reconstruct such 3D models, using multiple calibrated views of the object of interest.

A larger part of this thesis is then interested in *exemplar-based* recognition methods, using directly 2D example images for training, without any explicit 3D information. The appearance of objects is also defined as probability distributions of observable features, defined in a nonparametric manner through kernel density estimation, using image features from multiple training examples as supporting particles. The task of object localization is cast as the cross-correlation of distributions of features of the model and of the test image, which we efficiently solve through a voting-based algorithm. We then propose several techniques to perform *continuous* pose estimation, yielding a precision well beyond a mere classification among the discrete, trained viewpoints. One

of the proposed method in this regard consists in a generative model of appearance, capable of interpolating the appearance of learned objects (or object categories), which then allows optimizing explicitly for the pose of the object in the test image.

Our model of appearance, initially defined in general terms, is applied to the use of edge segments and of intensity gradients as image features. We are particularly interested in the use of gradients extracted at a coarse scale, and defined densely across images, as they can effectively represent shape as they capture the shading onto smooth non-textured surfaces. This allows handling some cases, common in robotic applications, of objects of primitive shapes with little texture and few discriminative details, which are challenging to recognize with most existing methods.

The proposed contributions, which all integrate seamlessly in a same coherent framework, proved successful on a number of tasks and datasets. Most interestingly, we obtain performance on well-studied tasks of localization in clutter and pose estimation, well above baseline methods, often on par with or superior to state-of-the-art method individually designed for each of those specific tasks, whereas the proposed framework is similarly applied to a wide range of problems.

Résumé

Le sujet de la reconnaissance d'objets est un problème central dans le domaine de la vision par ordinateur. En plus d'être étudié comme problème scientifique en tant que tel, il en découle également nombre d'applications pratiques. Nous nous intéressons ici aux applications robotiques telles que la manipulation et la saisie d'objets, dans les situations typiques que pourrait rencontrer un robot d'aide domestique. La reconnaissance visuelle d'objets dans ce contexte est alors cruciale pour permettre à la machine de comprendre son environnement, de localiser les objets et, finalement, de planifier des tâches comme par exemple leur saisie.

Cette thèse présente une série de méthodes qui s'appliquent à la détection, la localisation, la reconnaissance et l'estimation de pose, dans des images 2D, d'objets spécifiques et de catégories d'objets. Les techniques que nous proposons ont la particularité d'être applicables de manière générale, en considérant toutes ces tâches comme différentes facettes d'un même problème, ainsi qu'en évitant de nous focaliser sur un type particulier de caractéristiques d'images. Dans un premier temps, nous nous intéressons à l'utilisation de modèles 3D des objets, afin de faire une estimation de pose continue de ces objets. Ces modèles sont construits sur base de constellations de points, correspondant à des caractéristiques visuelles potentiellement observables, qui servent à définir une distribution de probabilités de ces caractéristiques en 3D. Cette distribution peut être projetée sur le plan image, et la tâche d'estimation de pose revient alors à maximiser la correspondance entre cette projection et l'image de la scène à analyser. En appliquant cette méthode aux bords comme caractéristiques visuelles, nous pouvons traiter des objets sans texture, et la formulation probabiliste fournit un moyen élégant de modéliser l'incertitude dans la définition des modèles. Ceux-ci peuvent ainsi être appris à partir d'observations plutôt qu'à partir de plans précisément dessinés a la main. Nous proposons par ailleurs une méthode permettant de reconstruire de tels modèles 3D à partir d'une série d'images calibrées d'un objet.

La plus grande partie de cette thèse se focalise ensuite sur la reconnaissance d'objets à base d'*images-exemples*, c'est-à-dire en utilisant des images 2D des objets comme données d'apprentissage, sans passer par une reconstruction 3D explicite. L'apparence d'un objet est représentée par une distribution de probabilités de caractéristiques observables (en 2D cette fois), que nous définissons de façon non paramétrique par estimation de densité par noyau. La tâche de la localisation d'objets est formulée comme la maximisation de la corrélation croisée entre les distributions représentant le modèle et l'image de test. Nous proposons aussi différents moyens d'estimer la pose *continue*

des objets, avec une précision bien au-delà d'une simple classification parmi les vues discrètes d'apprentissage. Une des méthodes proposées dans cette optique consiste en un modèle génératif, qui peut interpoler l'apparence de l'objet sous des points-de-vue quelconques, et permet ainsi d'optimiser explicitement la pose de l'objet dans l'image.

Notre modèle d'apparence est défini en termes généraux en ce qui concerne le type de caractéristiques d'images. Nous l'appliquons à l'utilisation de bords ainsi que de gradients d'intensité. L'utilisation de gradients extraits à échelle grossière dans les images est particulièrement intéressante car ceux-ci fournissent d'utiles indices liés à la forme de surfaces. Ceci permet de gérer des objets non texturés, ou avec peu de détails visuels, courants dans des applications robotiques, qui sont difficiles à reconnaitre avec les méthodes habituelles, utilisant uniquement les bords, par exemple.

L'ensemble des contributions proposées s'intègrent dans une formulation commune, et la méthode globale résultante a été évaluée sur un ensemble de tâches et de jeux de données. Nous obtenons des performances sur les tâches de localisation et d'estimation de pose bien supérieures aux méthodes de bases, et souvent comparables voir supérieures à l'état-de-l'art sur chaque tâche spécifique, alors que le système que nous proposons s'applique de façon similaire a un large éventail de tâches.

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Justus Piater. Not only did he get me interested in computer vision, while I was still a Master student, thanks to his lively and enthusiastic teaching. He also set me on exciting research avenues during my PhD, and provided a right balance of guidance and research freedom all along the years of work that lead to the completion of this thesis. I must also acknowledge the chance he offered me to move to such an exciting place as the University of Innsbruck after a first year at the University of Liège.

I also thank Prof. Jacques G. Verly, once the advisor for my Master thesis, and now the second supervisor of my PhD thesis. His devoted help to any request has always been much appreciated, and his thorough proof-reading helped polishing the final version of the present document.

I thank the other members of the Jury of this thesis, Prof. Aleš Leonardis, Prof. Jim Little, and Prof. Louis Wehenkel, for the time taken reading and evaluating my work.

Among my former colleagues at the Montefiore Institute, I am particularly debtful to Renaud Detry, for his availability for advice, and for the great technical ideas he developed in his own research, upon which my research initially built.

I thank all the people of the growing team of the "Intelligent and Interactive Systems" at the University of Innsbruck: Alex, Antonio, Emre, Hanchen, Heiko, Mirela, Sandor. The diversity of their backgrounds and wealth of knowledge everyone brought to the group made it a very stimulating and interesting place to work at. I must especially thank my friend Thomas who helped making a corner of our Austrian office a piece of Belgium far away from home, and contributed making every day at the office a fun day. I am also especially grateful to our lab technician, Simon, for his availability and abilities to solve any practical problem, whether with a flare-nut wrench, the right piece of paperwork, a soldering iron or the appropriate shell script.

This thesis would not have been possible without the financial support of the Belgian National Fund for Scientific Research (FNRS) and of the European Union, respectively thanks to a Research Fellowship (*Mandat d'Aspirant*, in French) and through the XPerience and IntellAct projects.

Finally, my greatest gratitude goes to my Belgian friends, my sister and my parents, for their care and support during those parts of the life away from the office. I cannot thank my parents enough for giving me the opportunity to pursue my interests, academic and otherwise, and for teaching me the value and merit of perseverance and hard work, without which none of this would have happened.

x

Contents

Chapter 1

Introduction

The field of Computer Vision is a fascinating research area, but which may prove frustrating to work in. Simple tasks for a human, such as recognizing an object in a photograph as a car or a plane, or locating a coffee mug in the picture of a kitchen scene, proves incredibly challenging for a computer. Although any person is capable of performing those tasks effortlessly, what remains less obvious is *how* our brain can do so. Therefore, the "holy grail" of computer vision, which would be the method capable of a thorough understanding of a complete scene from a single image, may still remain an elusive goal for some time, despite being a lively subject of research since the early years of computer science. In addition to being a tremendously interesting topic for scientific research in its own right, computer vision counts many practical applications. To name a few: automatic surveillance, monitoring of industrial processes, navigation of mobile robots, automatic indexing of image databases, primary sensing for domestic service robots, etc. Among those, we are particularly interested in the applications where the machine is ultimately expected to *interact* with its environment, and where the perception capabilities offered by computer vision will just constitute one piece of the puzzle, albeit a crucial one. This thesis therefore presents the results of research geared towards a central challenge of computer vision, broadly called object recognition, keeping in mind some particular constraints related to the tasks of manipulation and grasping of objects (Fig. 1.1). As we will see, this imposes additional challenges, but also brings particular opportunities, which lead to different approaches and design choices, compared to computer vision methods designed in a more general sense.

1.1 Current landscape of computer vision

The topic of object recognition has been a central one since the beginnings of computer vision. It actually encompasses a number of practical tasks, as will be detailed below. Considering computer vision as a self-standing discipline encourages the design of methods around particular techniques or methodologies, and not driven by end applications. This leads to the constant contributions of innovative techniques

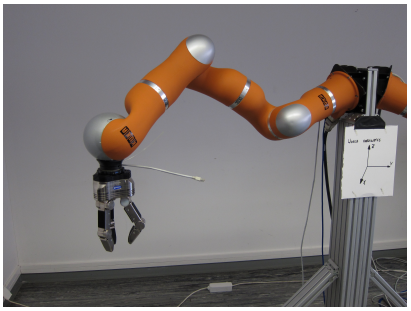
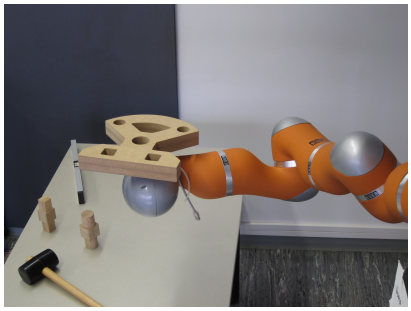



Figure 1.1: Scenario of robotic manipulation, where a robot arm is expected to pick up, move, and assemble objects.

and methods, but it also means that the application of these methods to practical problems may sometimes be trailing behind. In addition, a common practice in the field of computer vision for evaluating novel methods or algorithms, is to primarily consider well-established "benchmark" datasets. Those datasets are usually targeted at some very specific tasks, and can naturally represent only a precise set of conditions. The advantage of this common practice is to make competing algorithms directly comparable with each other, and has allowed the state of the art to reach very high levels of performance on some specific tasks. This however also brings a drawback, which is to push more developments on those specific tasks, while not encouraging developments in vastly different directions. It does actually lead to a lack of methods applicable to some particular conditions not well represented within those common datasets, but which we are nonetheless interested in, in the context of this thesis. For example, on the task of object recognition, methods are frequently proposed and designed around a single type of image information, image edges for example, or a specific type of image descriptor, histograms of intensity gradients for example. This example of focusing on image *edges*, and on the silhouette of objects is particularly frequent, and has led to the development of the entire field of research of *contour matching*. The state of the art reached near-perfect performance of datasets particularly suited to these methods — such as the "ETHZ Shape dataset" for object class detection — [9–11], but it is not always obvious how such methods would apply and perform in different conditions. A second side-effect of the use of benchmark datasets for primary evaluation is that different tasks represented by different datasets are then sometimes considered in isolation, independently of each other. For example, the tasks of shape detection (e.g. identifying bottles in an image), which are often evaluated on the afore-mentioned "ETHZ Shape dataset", is rarely considered at the same time as the task of viewpoint classification (e.g. identifying cars seen from the side versus cars seen from the front), commonly evaluated on the "3D Object dataset" [12]. However, both problems unarguably present common traits that may benefit from a similar treatment.

1.2 Problem statement

The methods proposed in this thesis try to alleviate the frequent, but undesirable practices discussed above in the four following ways. First, we propose methods that do not focus on the use of specific types of image features but which are equally suited to sparse image features (keypoints for examples) and dense image features (intensity gradients for example). Second, we propose methods that target both the recognition of specific objects (e.g. a particular car of a certain brand and specific color) and of object *categories* (any car in the general sense). The method that we will present in Chapter 8 is actually applicable to both cases identically, and can be at choice trained with data from a specific object or from different objects defining a category. Third, we consider all types of objects, from complex textured objects, cars for example, often encountered in the classical datasets for object recognition and well suited to classical image features and descriptors (as in [13] for example), but also non-textured objects of simple shapes, plates and kitchen knives for example, often encountered in the practical scenarios of robotic interaction that we are interested in. Such objects may seem simple in appearance, but their lack of distinctive visual characteristic make them difficult to identify in images, and are often simply avoided in the evaluation of existing methods, in profit of heavily textured objects, that can often be identified much more reliably (Fig. 1.2). Finally, fourth, we consider all following tasks related sides of a same problem. We thus try, to the possible extent, to tackle them — then to evaluate our performance on them — in a unified manner. These tasks consist in:

- **Localization** The goal is to identify the parts of the test image that belong to the object of interest, versus the parts of the image that correspond to background clutter. The result of localization is typically a set of bounding boxes, which encircle candidate objects in the image, each accompanied with a detection score.
- **Detection** One must decide whether the object of interest appears in the test image or not. One usually does this together with localization (both terms being then used interchangeably), by setting a threshold on scores of localizations to obtain a binary result for detections.
- **Classification among objects** One must determine which object among learned ones appears in the image. This typically involves learning discriminative classifiers, and may take into account contextual information within the image such as the background. The object model used for this task may not include any spatial structure, e.g. a "bag-of-words" models, and the results of classification thus does not necessarily involve the localization in the image of the object of interest. Such an approach would obviously not be satisfactory in the applications considered here.
- **Classification among discrete viewpoints (poses)** One must determine in which pose, or orientation, an object appears in, among learned ones (e.g. cars seen from the side and from the front). This has obvious relations with the task of recognizing an object versus another. This task bears different names: "pose estimation" is common in the field of robotics, while "viewpoint recognition" is more often used in the context of pure computer vision.

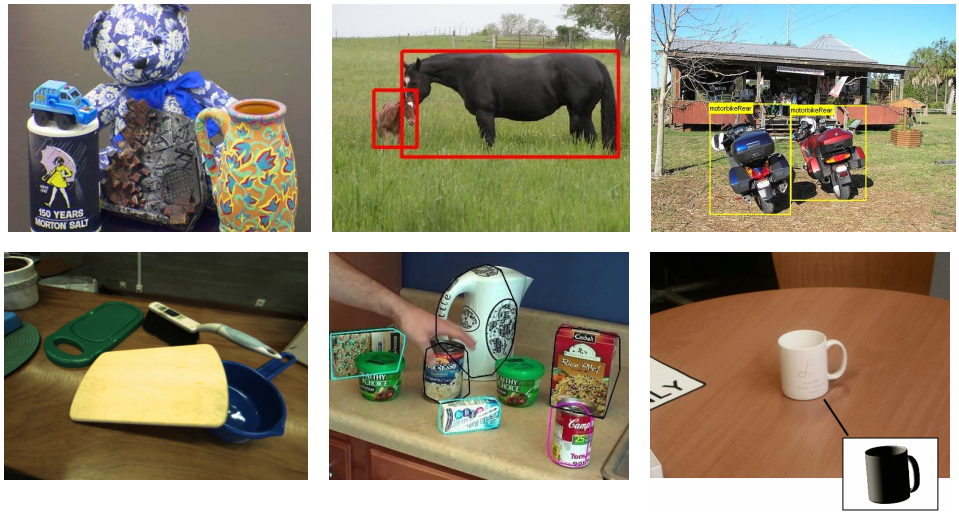


Figure 1.2: The task of object recognition is viewed somewhat differently in the fields of pure computer vision (first row) and robotics (second row). The ultimate goal in both cases is the high-level interpretation of complete scenes. In the former, lots of existing work focus on the modeling of the appearance of complex categories such as horses (top center). Common applications in robotics often involve objects of different nature and of smaller scale. Robotic applications also often involve more than a "bounding box" localization, and require accurate segmentation and/or 3D pose estimation (bottom center, bottom right). In both fields of computer vision and robotics, it is common to consider heavily textured objects: they are both frequent in the real world and easier to recognize thanks to strong visual characteristics (top left, bottom center). Sources: [13], [14].

- **Continuous viewpoint (pose) estimation** In this more challenging task, one must determine the precise orientation of the object (e.g. as Euler angles), with a resolution much higher than a "side" or "front" view. This task is seldom considered in the general field of computer vision, and has rather been studied by roboticists, since it often is a prerequisite for the robotic tasks of manipulation and grasping.

1.3 Object recognition in the communities of computer vision and robotics

We acknowledge the resemblance between the different tasks described above, which we broadly regroup under the term of "object recognition". We will first review how these tasks have been treated in the field of pure computer vision. We will then see how the particular needs within the robotics community have shaped different solutions. Note that this section is only a mere overview of the areas of research in which this thesis is inscribed. An appropriate and thorough literature review related to each specific topic will be presented in the beginning of each subsequent chapter.

1.3.1 Object recognition as a self-contained task in computer vision

The work on the topic of object recognition in the field of pure computer vision is most easily classified by "techniques", whereas the application-oriented view in robotics make them easier to classify by "goal". Early work in the vision community historically considered the appearance of an object in an image as a whole. Such classical techniques then involved, for example, performing principal component analysis on several training images of an object, in order to extract its most relevant visual characteristics. Other approaches extracted so-called "'global features" describing entire regions of the image corresponding to the whole object. The limitations of such techniques were a poor robustness to occlusions and deformations. This led to the now common paradigm of "local features", where images are first described by sets of features, which describe different parts of an image. These features are then handled, in a second step by a decision process that performs recognition, localization in the image or any related task. Generally, this separation conveniently serves two separate purposes. On the one hand, the description of the image with features or descriptors provides invariance to low-level variations, such as image noise, lighting, translation and scaling in the image. On the other hand, the decision process of the second step is then responsible, when performing recognition, for handling the more complex issues of intra-class variations of appearance, the presence of clutter in the image, the possible deformations of an object, etc.

Contributions to the field of object recognition can be related to one or both of the two steps introduced above. On the topic of image features, developments over the years have seen the introduction of sparse features, such as corners [15], key points [16], or regions [17], "semi-sparse" features, such as edges [18, 19], and dense features, such as gradient fields or convolution-based responses to different types of filters. Those features are then often associated to local appearance descriptors of image patches, and/or summarized into descriptors such as histograms [20–22]. At the other level of the decision process, common methods for recognition include rigid matching with sliding windows [23–26], the generalized Hough transform [20, 27, 28], or non-rigid matching with contour matching [11, 29, 30] or part-based models [31–33]. Modern approaches often include machine learning techniques, such as the learning of discriminative classifiers, in order to differentiate parts of the image belonging to background clutter versus those corresponding to the object of interest. Different techniques have been proposed to tackle the problem of recognition of both specific objects and object categories. The latter is however considered more challenging, due to the intra-class variations of appearance that must be dealt with. Indeed, a set of image features and descriptors as mentioned above (e.g. SIFT features, to name the most popular) can be descriptive and discriminative enough to identify a specific object, the cover of a specific book, for example. It will however require a more complex machinery to recognize *any* book after learning the appearance of a few examples. This is where machine learning methods, such as classifiers based on support vector machines (SVMs) [34] or AdaBoost [35] have recently proven very successful.

1.3.2 Object recognition as a tool in robotics

Methods for object recognition are seen from the field of robotics as tools that must serve some specific purpose. In the application of robotic manipulation and grasping, the output of interest is the actual 3D position and orientation (their combination being referred to as the pose) in the world of the objects to handle. Although this task is intrinsically similar to the one of localizing and encircling the object in the image with a bounding box (the output of many methods in computer vision), the needs of robotic applications lead to slightly different conventions, requirements and opportunities.

On the one hand, robotic applications impose particular needs, like the practical form of the result of the localization of the object, as mentioned above. While a classical recognition method will usually be limited to delineating the area of the input image that belongs to the object of interest, we will rather be here interested in recovering the actual 3D position of this object in the world, in the form of Euclidean coordinates. As discussed in more details in Section 1.5.1, the conversion from one to the other is however trivial given that the camera has been calibrated. Other practical matters of interest include robustness and computational requirements, in terms of resources and execution times (latency).

On the other hand, robotic applications may bring opportunities in the form of prior knowledge that may be used advantageously, to aid in the task of recognition. In particular, the robot is likely to know in which environment it is evolving, and thus which objects can appear in a scene and must be recognized. A gardening robot may not need to look for kitchen knives, and a robot evolving in an industrial environment probably does not need to recognize coffee mugs. The methods proposed in this thesis therefore focus on the recognition of one individual object at a time, rather than on the handling of large collections of different types of objects. Methods suited to such big databases have their place however, for example in the applications of content-based image retrieval. Another form of prior knowledge in the context of robotic applications comes from the actual imaging conditions. The calibration of the internal parameters of the camera (the focal length for example) can be assumed to be known, as well as the (approximate) distance between the camera and the captured scene. This information indicates at which scale the object of interest appears, and must be looked for in the image.

Finally, the application-driven approach in robotics means that the techniques are often used on an opportunistic basis. Requirements for speed, robustness and accuracy in pose estimation often pushes the use of the simple paradigm of rigid objects described by local invariant features, e.g. SIFT descriptors. The recognition process then typically involves finding specific correspondences between the image of the scene and the stored appearance model of the object. This applies well to specific objects, but unfortunately does not extend to categories of objects with similar reliability. For this reason, experiments with robotic systems are sometimes restricted to specific, known objects, instead of a more realistic use of object categories. These objects are moreover adequately chosen with distinctive texture and visual characteristics (Fig. 1.2), so that they can easily be handled by these simple recognition methods based on discriminative matching of visual descriptors. Another consequence of this opportunistic

approach is that there are no preset restrictions to specific training conditions. In particular, resorting to a 3D model of the object is common when it is necessary to perform continuous pose estimation. This indeed render the task much easier than with only plain 2D views as training data (although this is also a possibility, as we propose in Chapters 6 and 8). Similarly to the use of 3D data for *training*, the use of 3D information of the *test scene* is also a common way to render the perception and recognition tasks easier. These additional 3D observations can come, for example, from stereo vision or range scanners such as Kinect-style sensors. The use of such data (3D observations) is however an entire research avenue on its own, that we do not consider within this thesis. Any how much this extra information may help perception tasks, we believe that there is still much to be improved using 2D visual information alone. Another unique possibility offered within robotic applications is to interact with the world, either by moving and manipulating the observed objects [36, 37], or through active vision by moving the camera to appropriate viewpoints around the scene [38–40]. This also constitutes an entire field of research of its own, that we only touch upon in Chapter 6, with a method that uses several views of a scene (and thus parallax information) to disambiguate similar-looking poses.

1.4 Overview of contributions

The contributions presented within this thesis consist in a set of methods that address the problems of object recognition and pose estimation in 2D images, both at the level of specific objects and object categories. Our objective was to alleviate some of the common issues encountered with existing methods, as discussed above, by developing new methods applicable to various types of image features, various types of objects — including ones without texture or strong visual characteristics — and by considering the tasks of localization, recognition, and pose estimation as a single problem.

While the work presented here is definitely inscribed within the field of computer vision, our developments were initially motivated by the robotic applications of manipulation and grasping. As a consequence of this "application-driven" approach, the first chapters of this thesis (dealing with 3D models and pose estimation primarily) are somewhat influenced by the field of robotics. This is reflected by some design choices, e.g. by the representation of the object pose as quaternions, by some evaluation methods, and also simply by the scope of the literature review presented in those first chapters. In the later parts of the thesis however, the emphasis is brought back on computer vision, since most comparable work (on appearance-based recognition) is then clearly part of this field, and since our methods are then proving competitive also on more mainstream tasks and datasets used in pure computer vision.

We give below an overview of the proposed methods (Fig. 1.3). Each major contribution was originally presented in separate papers at various international conferences [1–8]. This thesis is formatted as a collection of those papers, each forming a self-standing chapter. As a consequence, the appropriate literature review related to each contribution is thus included in the beginning of each paper.

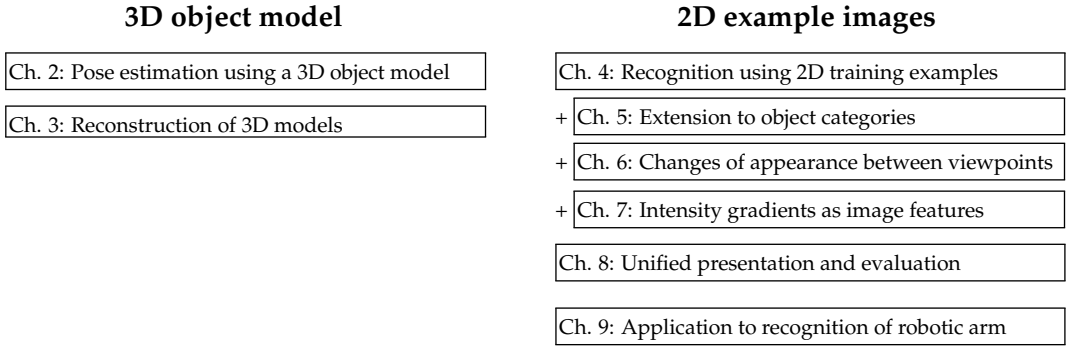


Figure 1.3: Overview of this thesis. Each significant contribution is presented as a self-standing chapter.

Considering the specific goal of estimating the precise pose of a known object in a 2D image, our first contribution, proposed in **Chapter 2**, uses a 3D model of the object, of which we minimize the reprojection error onto the test image. This method is built on existing work previously developed in our research group [41]. This work consists in the representation of objects as probabilistic 3D models, which are interestingly applicable to various types of image features. The models correspond to constellations of points, the geometric arrangement of which conveys information about object shape. All those points correspond to features potentially observable in 2D images of the object. The overall probabilistic formulation of this model serves to encode the uncertainty present in the description of both the appearance and the spatial arrangement of the features. This proved particularly appropriate, since our goal was to use object models learned trough observations, rather than man-made 3D CAD models. Such learned models obviously present imperfections, because of sensor noise to name just one example, and are thus more challenging to use than the typical, perfectly defined 3D CAD models. We applied the overall approach to the use of edge segments, where the "points" constituting the model correspond to points along the edges of the object. We thereby obtained very interesting results on the task of pose estimation. In particular, the use of edges allowed us to handle non-textured objects, which was one of the original goals.

In order to obtain the 3D object models needed by the pose estimation method introduced above, we initially used the sparse-stereo method of Krüger *et al.*, which reconstructs 3D edges from stereo imagery [42, 43]. To eliminate the reliance on that external piece of software, we designed our own, alternative method to obtain such reconstructions, specifically adapted to the conditions and type of data we were interested in. This method is presented in **Chapter 3**, and is suited for the scenario where a robotic agent would be free to observe the object to learn, with a single camera, but from different viewpoints (the camera being mounted on a robotic arm for example). A number of registered 2D views of the object are thus available, and serve to "back-project" these views to the 3D world. The triangulation of the observations from each input view define a probability distribution in the 3D world, which corresponds to the distribution of features (points along edges for example) that make up our object of in-

terest. This probabilistic treatment matches well with the intended use of the resulting models within the pose estimation method introduced above. Thanks to this method, we could obtain very clean 3D object models (with little noise or irrelevant clutter), made of continuous edge segments, particularly well suited to non-textured objects, which common methods for stereo reconstruction may struggle with. The integration of the two methods for (1) reconstructing models (Chapter 3) and (2) using them for pose estimation in 2D images (Chapter 2) is a project that is currently carried out by a student at the University of Innsbruck as his Master thesis, but which has not been completed yet.

While Chapters 2 and 3 were using explicit 3D models for internally representing the objects, the rest of the work in this thesis focuses on the use of 2D examples alone for training. The goal remains to recognize objects in a single image, but the data used to learn the appearance of those objects is now also made of 2D images exclusively. The motivations for this so-called "exemplar-based" approach are multiple. On the one hand, beyond the elegant simplicity of avoiding the reconstruction of a 3D model, some studies have shown that the human visual system may work in such a "lookup" manner for small ranges of viewpoints around previously-seen views [51], although this is still debated. On another hand, a practical advantage is that the training views do not need to all be initially available, and may be incrementally added to the model. Our method is initially presented **Chapter 4**, and then extended in several directions through the subsequent chapters. The initial formulation considers specific objects. In contrast with other existing methods, our method does not rely on establishing correspondences between image features of the learned model and of the test scene, and is thus applicable to similar-looking, non-distinctive features. We initially apply it, again, to the use of edges (points along edges, more precisely), but one specific aim we were targeting from the start was to use densely-defined image information, such as intensity gradients extracted at a coarse scale across images. Such gradients provide strong clues related to the shading on smooth surfaces (and thus shape), which is an often overlooked source of information, but of particular interest for handling objects of smooth shapes and non-textured surfaces. We finally reached this objective with the contributions presented in the latter parts of the thesis, in Chapters 7 and 8.

The method for exemplar-based recognition is extended from specific objects to object categories in **Chapter 5**. In addition to the recognition of specific, trained views, the capability of *continuous* pose estimation is provided by, basically, averaging the similarity of the test view with different, trained viewpoints. Although this technique is remarkably effective in practice, we found it unsatisfactory conceptually, and later proposed a more direct approach, presented in **Chapter 6**. In this alternative method, we explicitly encode how the appearance of the object varies between the discrete, trained viewpoints. This results in a generative model, capable of interpolating the appearance of the object at any, possibly unseen, viewpoint. We then use this model to optimize the pose of the object detected in the test view. Another, seemingly unrelated contribution is also presented in **Chapter 6**. It uses additional test views and the parallax between them to disambiguate poses of similar appearance. The reason for presenting these two contributions in a same chapter is the large overlap in the technical requirements (related to their implementation) for both methods.

As mentioned above, we also apply our model of appearance to another type of image features: intensity gradients, extracted at a coarse scale across images. **Chapter 7** examines how to use such information, in particular in relation with the issues of invariance to lighting conditions. We demonstrate the advantage brought by these image features with experiments, first on a set of particular objects that cannot be recognized reliably using edges alone, and then also on more general datasets, for which significant improvements are also brought by the use of these gradients in addition to edges alone.

A synthesis and summary of the most successful contributions of Chapters 4–7 is then given in **Chapter 8**, in the form of a journal paper (currently under review). This chapter presents these different contributions in a common formulation. Most interestingly, it also includes extensive evaluation on a number of benchmark datasets and on a number of tasks, i.e. shape detection, detection in clutter of objects and object categories, and discrete and continuous pose estimation. It demonstrates results on par with, or superior to, state-of-the-art methods on several on these datasets.

Finally, we include, in **Chapter 9**, preliminary results on a practical application of our recognition method. In a context of robotic manipulation, we apply it here to the recognition, not of the manipulated object, but of the manipulator itself, i.e. the robot arm. The robot considered is made of parts of smooth shapes with little texture, and our recognition method is thus an excellent candidate for this challenging recognition task. We extend our method to handle such an articulated object, whereas the previous chapters were assuming rigid objects. This ultimately provides the capability of identifying, then segmenting out the robot arm from images of a camera monitoring a scene of a manipulation scenario.

1.5 Technical background

The following sections introduce some technical concepts and conventions that are shared by several chapters throughout this thesis.

1.5.1 Representation of object pose

In the context of object recognition of this thesis, we call the *pose* of an object its position in the 3D world together with its orientation. Other uses of the term exist in different contexts; for example, the task of human body pose estimation involves determining the relative configuration of the body parts of an articulated model of the human body. Object pose estimation is most often used in the context of robotic manipulation and grasping, where the resulting information is paramount to control the robot e.g. to move its gripper to an appropriate graspable part of the object. The pose is thus the combination of a 3D position and a 3D orientation, which present 6 degrees of freedom altogether. Assuming a reference frame has been defined, e.g. aligned with the camera, the position is most simply described as 3D coordinates $x \in \mathbb{R}^3$ in this frame. For the orientation (an element of the rotation group $SO(3)$), several options exist, such as

Euler angles, rotation matrices, and quaternions. In our early work (Chapter 2), we use quaternions to represent the 3D orientation. Unit-length quaternions form the 3-sphere S^3 (the set of unit vectors in $\mathbb{R}^4$), and offer practical advantages such as numerical stability and being free of singularities.

Most work on object recognition in the field of pure computer vision is not concerned with recovering such precise information about the object. The typical output is merely the localization of the object in the image, in the form of a bounding box around the object. However, the scale at which appears the object (relative to the training examples) is readily available from the size of this bounding box. Similarly, one usually also identifies a possible rotation — also relative to the training examples — in the image plane. In the case of "multiview" models, the object of interest is learned with example images taken from different viewpoints. The viewpoint, i.e. which "side" of the object is facing the camera, corresponds to a point on the 2-sphere ($v \in S^2$). What is then worth noting is that this information presents the same six degrees of freedom as the representations used in the field of robotics. Assuming the camera used has been calibrated (for internal parameters and for its own pose relative to the chosen coordinate system), the localization and the scale in the image of the detected object determine its 3D position in the world. Its scale in particular, together with the focal length of the camera, determine the actual distance between the camera and the object. The viewpoint, together with the in-plane rotation in the image, determine its 3D pose. The viewpoint is commonly parametrized by Euler angles of azimuth ($\phi \in [0, 2\pi[$) and elevation ($\theta \in [-\pi, \pi]$). This is the convention used in our later work (Chapters 6–8), since it is used in most existing datasets that include objects viewed under different viewpoints. A description of the viewpoint with azimuth and elevation angles is indeed very intuitive, and directly corresponds to the conditions in which such datasets are created, for example with the object placed on a turntable (varying the azimuth angle) and the camera being placed at a few fixed heights (elevation angles). Moreover, most of the existing datasets we used in those later chapters do not provide any calibration information of the camera used, which prevented the conversion from "image-space" to "world-space" representations. The choice of the representations of the pose used throughout the different chapters of this thesis are thus only a result of the conventions used by competing methods and by the existing datasets used for evaluation.

1.5.2 Representation of probability distributions and density estimation

A common trait to our methods is to represent both the observations of a test scene and the model learned of an object, as probability distributions of observable features modeled by probability density functions. This serves to encode imprecisions in the observations, e.g. from sensor noise, as well the inherent uncertainty in the learned models. These density functions are built from the actual image features observed in the test or training images, using the concept of density estimation. Those observed features, called "particles" in this context, can thus be thought of as random samples of the underlying density that we want to represent. Density estimation can be performed in a parametric or nonparametric manner. With parametric methods, the density is represented by a small number of relatively complex kernels, of which the

parameters are tuned in order to best fit the available set of particles (and represent the underlying density). A popular model of this type is the mixture of Gaussians, where the number, the mean, and the covariance of the kernels are tuned with the Expectation-Maximization algorithm [44, 45]. By contrast, nonparametric methods rely on large numbers of simple kernels, attached to each of the available particles, and which present few parameters that are generally set similarly for all of them. We use a technique of this second type, with Kernel Density Estimation (KDE) [46]. Simple kernels are assigned to all image features (particles), and the density in any region of the domain of these features is defined as the sum of these kernels. This allows modeling complex multimodal distributions without any prior assumptions of density shape and avoiding the problems of mixture fitting or of choosing the number of components of parametric methods.

The modeling of probability densities with KDE as described above is generally applicable to various types of image features, and only necessitates to define an appropriate kernel function for each type of feature. A recurrent example in our work is the use of "edge points", i.e. points identified along edges in the image. Such an edge point x is defined by its location in the image $x_i^{pos} \in \mathbb{R}^2$, and the local (tangent) orientation of the edge at this point, an angle $x_i^{ori} \in S^{1+} = [0, \pi[$. We want to define a kernel that encodes uncertainty on both the position and the appearance (orientation) of the features. We therefore define a kernel as a product of two elements:

$$\mathrm{K}(\,\cdot\,;\; x_i) \;=\; \mathcal{N}(\,\cdot\,; x_i^{pos}, \sigma)\;\; \mathrm{VM}^{+}(\,\cdot\,; x_i^{ori}, \kappa)\;. \tag{1.1}$$

The first factor $\mathcal{N}(\ldots)$ corresponds to a univariate Gaussian distribution for the position, assuming a simple isotropic spatial uncertainty, while the second corresponds to a von Mises distribution [47] (similar to a wrapped Gaussian) on $[0, \pi[$ for the orientation. From a set of image features x_i extracted from an image, kernel density estimation can then be used to estimate the probability density $d(x)$ at any point x, for example to evaluate the likelihood of observing a horizontal edge at a particular location in the image:

$$d(x) \;\simeq\; \frac{1}{n} \sum_{i}^{n} \mathrm{K}(x;\; x_i)\; .d(x) \;\simeq\; \frac{1}{n} \sum_{i}^{n} \mathrm{K}(x;\; x_i)\;. \tag{1.2}$$

Details concerning the use of different image features with this formulation will be discussed in subsequent chapters, as well as the specifics of the weighting of particles and of the sampling of densities defined through KDE.

The representation of images as probability densities of image features with the nonparametric formulation above involves using large number of particles, especially with our "edge points", and even more so with the dense features (intensity gradients, defined for each pixel of the image) that we use in Chapter 8. We will introduce techniques in subsequent chapters to make the manipulation of such data tractable, e.g. by performing object detection with a voting algorithm. In addition, a generally applicable procedure to reduce the computing requirements is to simply resample the set of particles. A random or systematic sampling could be appropriate, but, in pactice, we obtained more consistent results with the semi-random procedures illustrated in

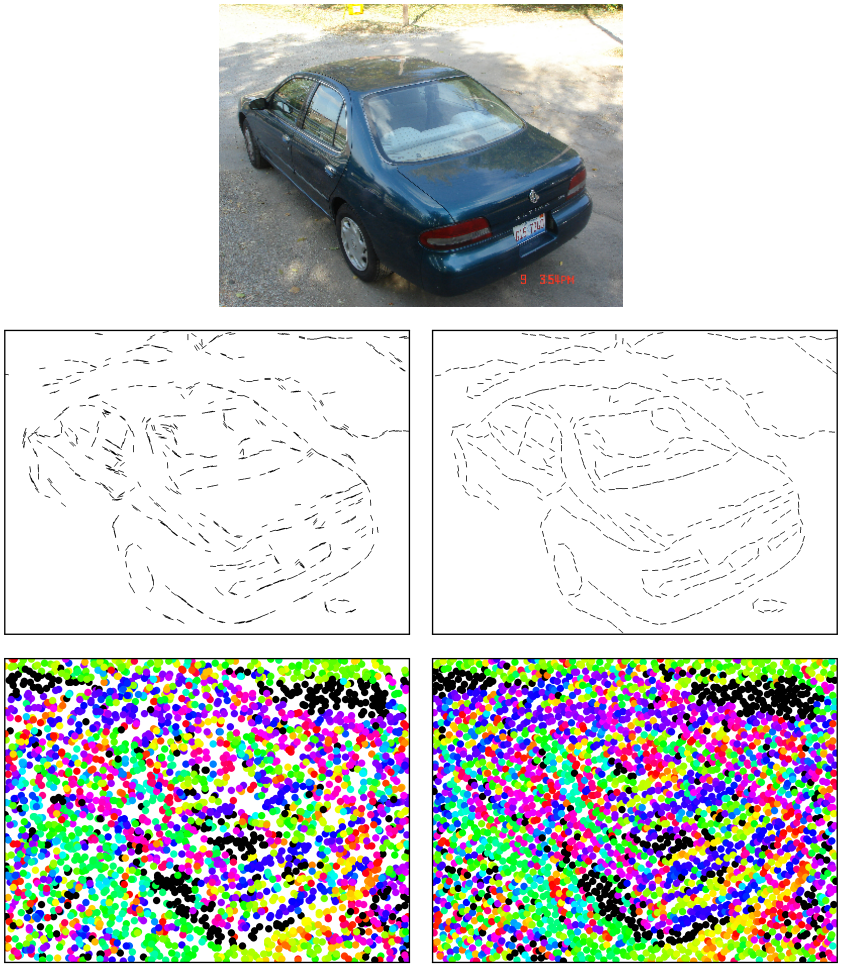


Figure 1.4: Comparison of random (left) and semirandom (right) procedures for sampling image features of a single test image (top). The two procedures are compared with a same number of samples. In the case of oriented edges points (middle row), the semirandom procedure selects samples along edges spaced with random *bounded* intervals. In the case of gradient points (bottom row), it selects samples on a grid built from a Halton sequence. The semirandom procedures ensure a better homogeneity of samples, in order to evenly represent all parts of the image, even when using small numbers of samples (which is desirable for efficiency).

Fig. 1.4. In the case of edge points, we follow edges and we select points on those edges at random *bounded* intervals (e.g. every 5 to 10 pixels). This ensures that small numbers of samples will cover the different edges in different parts of the image with a similar density. In the case of dense features such as the gradients mentioned above, instead of retaining all pixels of the image as image features, we select some of them on a grid built from a Halton sequence [48] over the image. This approach is typical with the so-called quasi-Monte Carlo methods; it also ensures that samples are distributed randomly but with similar local densities across the sampling space (i.e. across the whole image, here).

1.5.3 Monte Carlo methods

The use of the probability densities of image features introduced above commonly involves the computation of integrals. In particular, we often use integrals over products of densities of features as a measure of their visual similarity. We then use Monte Carlo methods to numerically evaluate the value of such integrals. Monte Carlo integration is based on a random exploration of the integration domain. Considering two density functions $f(x)$ and $g(x)$, defined on the same domain, we evaluate the integral of the product by drawing random samples from one, and by averaging the values of the other at those points [49]. Formally,

$$\int f(x)\, g(x) \mathrm{d}x \;\simeq\; \frac{1}{n} \sum_{\ell}^{n} f(x_\ell) \quad \text{with} \quad x_\ell \sim g(x)\ . \tag{1.3}$$

Similarly, we formulate the problem of object detection in the image as a cross-correlation. The goal is then to maximize the visual similarity of a template, represented by $g(x)$, with a test image, $f(x)$, by aligning it with rigid transformations represented by w (see Chapter 8 for a more thorough description) and applied by the function $t_w(x)$. The problem of detection then corresponds to the maximization of the cross-correlation

$$c(w) \;=\; \int f(x)\; g\big(t_w(x)\big)\; \mathrm{d}x\ , \tag{1.4}$$

which is evaluated similarly as in Eq. 1.3 and gives

$$c(w) \;\simeq\; \frac{1}{n} \sum_{\ell}^{n} f\big(t_w(x_\ell)\big) \quad \text{with} \quad x_\ell \sim g(x)\ . \tag{1.5}$$

As will be proposed in the next chapters, this maximization problem can also efficiently be solved with algorithms based on a voting procedure.

Chapter 2

Pose estimation using a 3D object model

The initial motivation for the work presented in this chapter comes from existing activities in our group on the topic of robotic grasping. The goal here is to perform pose estimation of known objects in 2D images, using provided 3D models of those objects. This explains some conventions used, for example for representing the pose in "world-space" coordinates (as opposed to "image-space" coordinates, as discussed in Section 1.5.1). We thus recover the pose of the object as a 3D position and a 3D orientation, in a Euclidean frame aligned, by convention, with the camera. Some results in the evaluation part of this chapter show reprojections of the object models superimposed onto the test images. This is provided for visualization purposes only, as the actual output of interest is the 6 degrees-of-freedom pose of the recognized object.

Technically, the method builds onto existing work of Detry *et al.* [41], which consists in probabilistic 3D object models. Such models encode both the geometry and the appearance of objects as a constellation of points that correspond to observable features. These points actually serve to represent a continuous probability distribution of such features in 3D, defined in a nonparametric manner using KDE (as introduced in Section 1.5.2). Such probabilistic object models were used originally [41] to perform pose estimation of known objects in a scene, using *3D* observations of the scene, obtained with stereo vision or with range scanners. The task of pose estimation, in that context, was thus to register, or "align", the 3D model with the 3D observations of the scene [50].

In our work, we adapted the principles of that existing method [41] to perform pose estimation with the same type of model but with only *2D* observations of the scene, i.e. a single image. The 2D observations extracted from the test image correspond to the same – potentially – observable features that make up the model, and the task of pose estimation is then to optimize the "match" between the reprojection of that 3D model onto the test image plane, and the test image itself. We reuse the original probabilistic formulation of the 3D model, and adopt a similar approach to turn the image features, extracted from the test image, into a continuous distribution of features.

The solution to the optimization of the pose is provided by a Monte Carlo algorithm that explores the pose space to identify local maxima of our objective function (the similarity between the test image and the reprojected model). The high dimensionality of the pose space (presenting 6 degrees of freedom) makes this challenging, but we obtained nevertheless good results as demonstrated by a number of experiments.

The initial choice of framing the task strictly as a pose estimation problem is responsible for a practical drawback: it limited the choice of datasets that we could use for the evaluation. Indeed, few existing datasets provide ground truth information of the pose of the objects, or even just the calibration of the camera (which would be required to retrieve the pose as 3D coordinates in a Euclidean coordinate system).

The paper included in the following pages was presented at the 2011 *German Conference on Pattern Recognition (DAGM)*.

Probabilistic Object Models for Pose Estimation in 2D Images

Damien Teney[1] and Justus Piater[2]

[1] University of Liège, Belgium
Damien.Teney@ULg.ac.at
[2] University of Innsbruck, Austria
Justus.Piater@UIBK.ac.at

Abstract. We present a novel way of performing pose estimation of known objects in 2D images. We follow a probabilistic approach for modeling objects and representing the observations. These object models are suited to various types of observable visual features, and are demonstrated here with edge segments. Even imperfect models, learned from single stereo views of objects, can be used to infer the maximum-likelihood pose of the object in a novel scene, using a Metropolis-Hastings MCMC algorithm, given a single, calibrated 2D view of the scene. The probabilistic approach does not require explicit model-to-scene correspondences, allowing the system to handle objects without individually-identifiable features. We demonstrate the suitability of these object models to pose estimation in 2D images through qualitative and quantitative evaluations, as we show that the pose of textureless objects can be recovered in scenes with clutter and occlusion.

1 Introduction

Estimating the 3D pose of a known object in a scene has many applications in different domains, such as robotic interaction and grasping [1,6,13], augmented reality [7,9,19] and the tracking of objects [11]. The observations of such a scene can sometimes be provided as a 3D reconstruction of the scene [4], e.g. through stereo vision [5]. However, in many scenarios, stereo reconstructions are unavailable or unreliable, due to resource limitations or to imaging conditions such as a lack of scene texture.

This paper addresses the use of a single, monocular image as the source of scene observations. Some methods in this context were proposed to make use of the appearance of the object as a whole [6,13,15]. These so-called *appearance-based* methods however suffer from the need of a large number of training views. The state-of-the-art methods in the domain rather rely on matching characteristic, local features between the observations of the scene and a stored, 3D model of the object [1,7,17]. This approach, although efficient with textured objects or otherwise matchable features, would fail when considering non-textured objects, or visual features that cannot be as precisely located as the texture patches or geometric features used in the classical methods. Hsiao et al.'s method [8] seeks

This article is part of Chapter 2: *Pose estimation using a 3D object model* (p.17)
It originally appeared in: *2011 German Conference on Pattern Recognition (DAGM)*

to better handle multiple possible correspondences between the model and scene features, but still requires a large fraction of exact matches to work efficiently.

The proposed method follows a similar approach to the aforementioned references for modeling the object as a 3D set of observable features, but it is different in the sense that few assumptions are made about the type of features used, and in that it does not rely on establishing specific matches between features of the model and features of the observed scene. For this purpose, we represent both the object model and the 2D observations of a scene as probabilistic distributions of visual features. The model is built from 3D observations that can be provided by any external, independent system. One of the main interests of the proposed method, in addition to the genericity of the underlying principles, is its ability to effectively handle non-textured objects. The general method itself does not make particular assumptions about the type of features used, except that they must have a given, although not necessarily exact, position in space, and they must be potentially observable in a 2D view of the object.

In order to demonstrate the capabilities of the proposed method at handling textureless objects, we apply it to the use of local edge segments as observations. Practically, such features cannot be precisely and reliably observed in 2D images, e.g., due the ambiguity arising from multiple close edges, 3D geometry such as rounded edges, or depth discontinuities that change with the point of view. Such problems motivate the probabilistic approach used to represent the scene observations.

The 3D observations used to build the model are provided by an external system that performs stereopsis on a single pair of images. Such a model can thus be quickly and automatically learned, at the expense of imprecision and imperfections in the model. This again motivates the use of a probabilistic distribution of features as the object model. Other *model-based* methods proposed in the literature have used rigid learned [7,17] or preprogrammed (CAD) models [9,19], but such CAD models are, in general, not available. Our approach for object modeling is more similar to the work of Detry et al. [5], where an object is modeled as a set of parts, themselves defined as probability distribution of smaller visual features. The main contribution of this paper is the extension of those principles to the use of 2D observations.

The representations of the object model and of the scene observations that we just introduced can then be used to perform pose estimation in monocular images, using an inference mechanism. Algorithms such as belief propagation [5] and Metropolis-Hastings MCMC methods [4] were proposed in the literature to solve similar problems, and we adapt the algorithm presented in that last reference to our specific type of model and observations.

Finally, our method provides a rigorous framework for integrating evidence from multiple views, yielding increased accuracy with only a linear increase of computation time with respect to the number of views. Using several views of a scene is implicitly accomplished when using a stereo pair of images, together with a method operating on 3D observations [5]. However, our approach does not seek matches between the two images, as stereopsis does, and can thus handle

This article is part of Chapter 2: *Pose estimation using a 3D object model* (p.18)
It originally appeared in: *2011 German Conference on Pattern Recognition (DAGM)*

arbitrarily wide baselines. Other methods for handling multiple views with a 2D method have been proposed [2,14]. In these methods however, the underlying process relies on the matching of characteristic features.

2 Object Model

Our object model is an extension of earlier work [4]. For completeness and clarity, the upcoming sections include essential background following this source.

2.1 General form

We use a 3D model that allows us to represent a probabilistic distribution of 3D features that compose the model. These features must be characterized by a localization in the 3D space, and can further be characterized by other observable characteristics, such as an orientation or an appearance descriptor. The model of an object is built using a set

$$M = \left\{ \left(\lambda^{\ell}, \alpha^{\ell} \right) \right\}_{\ell \in [1,n]} \tag{1}$$

of features, where $\lambda^{\ell} \in \mathbb{R}^3$ represents the location of a feature, and $\alpha^{\ell} \in \mathcal{A}$ is a (possibly zero-element) vector of its other characteristics from a predefined appearance space $\mathcal{A}$. When learning an object model, the set of features M is decomposed into q distinct subsets M_i, with $i \in [1, q]$, which correspond ideally to the different parts of the object. This step allows the pose estimation algorithm presented below to give equal importance to each of the parts, therefore avoiding distinctive but small parts being overwhelmed by larger sections of the object. The procedure used to identify such parts is detailed in [4].

Our method relies on a continuous probability distribution of 3D features to represent the model. Such a distribution can be built using Kernel Density Estimation (KDE), directly using the features of M_i as supporting particles [5,18]. To each feature of M_i is assigned a kernel function, the normalized sum of which yields a probability density function $\psi_i(x)$ defined on $\mathbb{R}^3 \times \mathcal{A}$. The kernels assigned to the features of M_i will depend on the type of these features.

Reusing the distribution of 3D features of part i, ψ_i, and considering an intrinsically calibrated camera, we now define $\psi'_{i,w}$ as the 2D projection onto the image plane of that distribution set into pose w, with $w \in SE(3)$, the group of 3D poses. Such a distribution is defined on the 2D appearance space, which corresponds to $\mathbb{R}^2 \times \mathcal{B}$, where $\mathcal{B}$ is the projected equivalent of $\mathcal{A}$. For example, if $\mathcal{A}$ is the space of 3D orientations, $\mathcal{B}$ would be the space of 2D orientations observable on an image. Similarly, if $\mathcal{A}$ is a projection-independent appearance space of 3D features, $\mathcal{B}$ would be the simple appearance space of direct 2D observations of such features.

Practically, $\psi'_{i,w}$ can be obtained by setting the features of M_i into pose w, and projecting them onto the image plane (Fig. 1c). The resulting 2D features $\in \mathbb{R}^2 \times \mathcal{B}$ can, similarly to the 3D points, be used as particles to support a KDE on that space, using an equivalent projection of the kernels used in 3D.

This article is part of Chapter 2: *Pose estimation using a 3D object model* (p.19)
It originally appeared in: *2011 German Conference on Pattern Recognition (DAGM)*

2.2 Use of edge segments

This paper presents the particular application of the object model presented above to the use of local edge segments as visual features. Those features basically correspond to 3D oriented points, which are characterized, in addition to their localization in 3D, by an orientation along a line in 3D. Therefore, reusing the notations introduced above, the space $\mathcal{A}$, on which the elements α^ℓ are defined, corresponds to the half 2-sphere S^2_+, i.e. half of the space of 3D unit vectors. The kernels used to compose a 3D probability distribution ψ_i can then be decomposed into a position and an orientation part [5,18]. The first is chosen to be a Gaussian trivariate isotropic distribution, and the latter a von Mises-Fisher distribution on S^2_+. The bandwidth of the position kernel is then set to a fraction of the size of the object, whereas the bandwidth of the orientation kernel is set to a constant. The 2D equivalent of those distributions are obtained using classical projection equations. Fig. 2 depicts the correspondence between the 2D and 3D forms of a particle corresponding to an edge segment and its associated kernel.

The visual features used in our implementation are provided by the external Early Cognitive Vision (ECV) system of Krüger et al. [12,16]. This system extracts, from a given image, oriented edge features in 2D, but can also process a stereo pair of images to give 3D oriented edge features we use to build object models (Fig. 1b).

3 Scene observations

The observations we can make of a scene are modeled as a probability distribution in a similar way to the model. The observations are given as a set

$$O = \left\{ \left(\delta^\ell, \beta^\ell \right) \right\}_{\ell \in [1,m]} \tag{2}$$

of features, where $\delta^\ell \in \mathbb{R}^2$ is the position of the feature on the image plane, and $\beta^\ell \in \mathcal{B}$ are its observable characteristics. These characteristics must obviously be a projected equivalent to those composing the object model. Here again, the features contained in O can directly be used as particles to support a continuous probability density, using KDE.

In the particular case of edge segments, the observations correspond to 2D oriented points (Fig. 1e). They are thus defined on $\mathbb{R}^2 \times \mathcal{B}$ with $\mathcal{B} = [0, \pi[$. As mentioned before, the uncertainty on the position and orientation of visual features like edge segments can arise from different sources, and no particular assumptions can thus be made on the shape of their probability distribution. The kernels used here are thus simple bivariate isotropic Gaussians for the position part, and a mixture of two antipodal von Mises distributions for the orientation part. The sum of those kernels, associated with each point of O, then yields a continuous probability density function $\phi(x)$ defined on $\mathbb{R}^2 \times [0, \pi[$ (Fig. 1f).

This article is part of Chapter 2: *Pose estimation using a 3D object model* (p.20)
It originally appeared in: *2011 German Conference on Pattern Recognition (DAGM)*

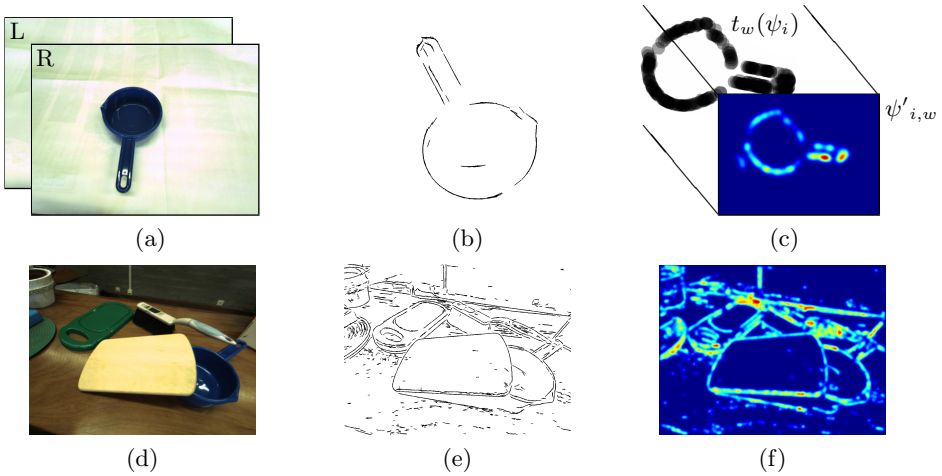


Fig. 1: Proposed method applied to edge segments (orientation of segments not represented). (a) Stereo images used to build object model; (b) 3D edge segments that compose the model; (c) probabilistic model (ψ_i) in pose w, spheres representing the position kernel (their size is set to one standard deviation), and its simulated projection in 2D ($\psi'_{i,w}$; blue and red represent resp. lowest and highest probability densities); (d) image of a scene; (e) 2D edge segments used as observations; (f) probabilistic representation of observations (ϕ).

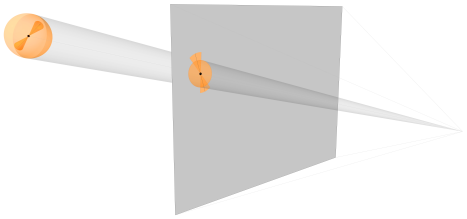


Fig. 2: Correspondence of 3D edge segment and associated kernel, with their 2D projection on image plane. Orange boundaries represent one standard deviation.

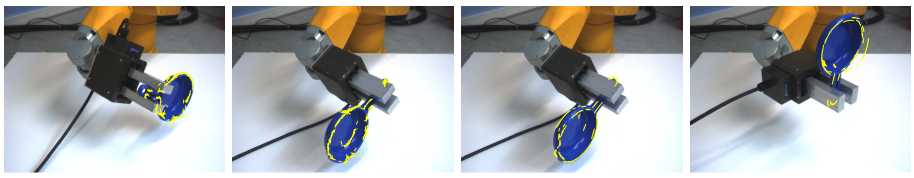


Fig. 3: Results of pose estimation; model features reprojected on input image. (a) Good result (close to ground truth); (b) good result; (c) same frame as (b) with incorrect result, orientation error of about 80°, even though the reprojection matches observations slightly better than (b); (d) incorrect result, insufficient observations extracted from pan bottom, and orientation error of about 180°.

This article is part of Chapter 2: *Pose estimation using a 3D object model* (p.21)
It originally appeared in: *2011 German Conference on Pattern Recognition (DAGM)*

4 Pose estimation

The object and observation models presented above allow us to estimate the pose of a known object in a cluttered scene. This process relies on the idea that the 2D, projected probability distribution of the 3D model defined above can be used as a "template" over the observations, so that one can easily measure the likelihood of a given pose.

Let us consider a known object, for which we have a model composed of q parts M_i ($i \in [1, q]$), which in turn define ψ_i and ${\psi'}_{i,w}$. On the other hand, we have a scene, defined by a set of observations O, leading to a probabilistic representation ϕ of that scene. We model the pose of the object in the scene with a random variable $W \in SE(3)$. The distribution of object poses in the scene is then given by

$$p(w) \propto \prod_{i=1}^{q} m_i(w) \;, \tag{3}$$

with $m_i(w)$ being the cross-correlation of the scene observations $\phi(x)$ with the projection ${\psi'}_{i,w}$ of the ith part of the model transformed into pose w, that is,

$$m_i(w) = \int_{\mathbb{R}^2 \times \mathcal{B}} {\psi'}_{i,w}(x)\, \phi(x)\, \mathrm{d}x \;. \tag{4}$$

Computing the maximum-likelihood object pose $\arg\max_w p(w)$, although analytically intractable, can be approximated using Monte Carlo methods. We extend the method proposed in [4], which computes the pose via simulated annealing on a Markov chain. The chain is defined with a mixture of local- and global-proposal Metropolis Hastings transition kernels. Simulated annealing does not guarantee convergence to the global maximum of $p(w)$, and we thus run several chains in parallel, and eventually select the best estimate. In practice, a strong prior is usually available concerning the distance between the camera and the object, e.g., as information on the scale at which the object can appear in an image. The global transition kernel can benefit from this prior to favor more likely proposals, and therefore drive the inference process more quickly towards the global optimum.

As mentioned above, the proposed method naturally extends to observations from v multiple views. We define $m_{i,j}(w)$ similarly to Eq. 4 but relative to specific views j, $j = 1, \ldots, v$. Accounting for observations from all available views, Eq. 3 then becomes

$$p(w) \propto \prod_{j=1}^{v} \prod_{i=1}^{q} m_{i,j}(w) \;, \tag{5}$$

which is handled by the inference process similarly to the single-view case.

5 Evaluation

This sections presents the applicability of the proposed method for estimating the pose of objects on two publicly available datasets [3,10].

This article is part of Chapter 2: *Pose estimation using a 3D object model* (p.22)
It originally appeared in: *2011 German Conference on Pattern Recognition (DAGM)*

5.1 Experimental setup

In this work, each model is built from one manually segmented stereo view of the object (such as Fig. 1a). The models used here are typically composed of between 1 and 4 parts, containing around 300 to 500 observations in total. Pose estimation is performed on single 1280×960 images taken with an intrinsically calibrated camera. The number of parallel inference processes (see Section 4) is set to 16. On a typical 8-core desktop computer, the pose estimation process on a single view typically takes about 20 to 30 seconds. Also, as proposed in Section 4 and detailed below, a crude estimate of the distance between the camera and the object is given as an input to the system.

The ECV observations we use (see Section 2.2) can be characterized with an appearance descriptor composed of the two colors found on the sides of the edge. This appearance information does not enter into the inference procedure. However, in the following experiments we use it to discard those scene observations whose colors do not match any of the model features. This step, although not mandatory, helps the pose estimation process to converge more quickly to the globally best result by limiting the number of local optima.

5.2 Rotating object

We first evaluated our method on a sequence showing a plastic pan undergoing a rotation of 360° in the gripper of a robotic arm [10]. The ground truth motion of the object in the 36 frames of the sequence is thus known. The estimate of the distance to the object, given as input to the system, is the same for the whole sequence, and is a rough estimate of the distance between the gripper and the camera (about 700 mm). Let us note that, for some images of the sequence, this estimate is actually quite different from the exact object-camera distance, since the object is not rotating exactly around its center.

This publicly available dataset is composed of stereo images, and we used the frame corresponding to a rotation of 50° to learn the model, as it gives a good overall view of the object. Four types of experiments were then performed (Fig. 4). First, the pose of the object was estimated in each frame of the sequence, using one single view. One can observe that correct pose estimates can mostly be made close to the viewpoint used for learning the model (Fig. 4). A number of results have an orientation error of almost 180°, which correspond to a special case (Fig. 3d) that can be explained by the flat and almost symmetrical object we consider. Indeed, if very few observations are extracted from the bottom of the pan, only the handle and the top rim of the object can be matched to the image. Another large number of incorrect pose estimates have orientation errors of 70–110°; most of them correspond to ambiguities inherent to a 2D projection, as illustrated on Fig. 3b–c. Similarly, most of the translation errors occur along the camera-object axis, as an inherent limitation of 2D observations. The percentage of correct pose estimates, defined by orientation and translation errors of less than 10° and 30 mm resp., and evaluated over the whole sequence, is only 20%. Second, the same experiment is performed using two views. Some

This article is part of Chapter 2: *Pose estimation using a 3D object model* (p.23)
It originally appeared in: *2011 German Conference on Pattern Recognition (DAGM)*

of the ambiguities can then be resolved, and this percentage rises to 60%. This result can be compared to the evaluation of Detry et al. [5] on a similar sequence, which achieved a score of only 40–50%. We stress that the latter method relied on 3D observations computed from stereo, whereas our method uses one or more 2D images directly, and is not limited to short-baseline stereo pairs.

Finally, we used our framework to track the pose of the object over the whole sequence, using one and two views, respectively. The pose is initialized with ground truth information for the first frame, and is then tracked from one frame to the next, using the same process as outlined in Section 4, but without the use of global proposals in the chain, and thus limiting the inference process to a local search. These experiments yield very good results (see Fig. 4), the remaining error being mostly due to the limitations of the model, learned from a single view of the object.

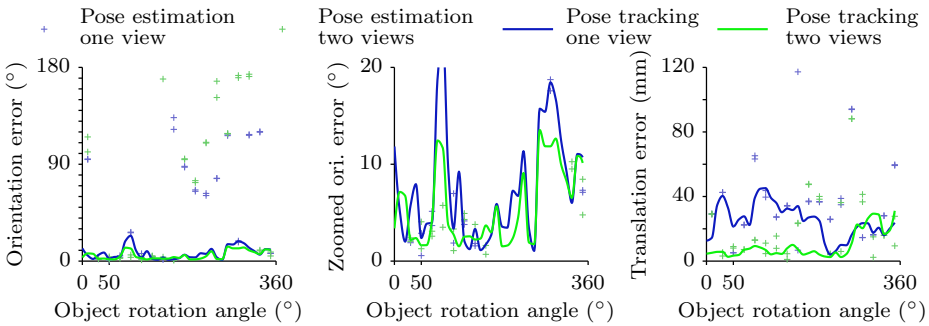


Fig. 4: Results of the "rotating object" sequence. For pose estimation, one marker represents one run of the algorithm (the same number of runs are executed for each frame). For pose tracking, the lines represent means over multiple runs.

5.3 Cluttered scenes

We evaluated the robustness of our method to clutter and occlusions by computing the pose of various objects in several cluttered scenes [3], using a single input image. The estimate of the distance to the objects, used as input, is the same for all scenes and objects, and roughly corresponds to the distance between the camera and the table on which the objects are placed (about 370 mm). Here again, this is an only crude estimate, as the actual distance to the objects varies from 200 to 600 mm.

Several of these scenes are presented in Fig. 5, with object models superimposed in the estimated pose. Sometimes, insufficient observations are extracted from the image, and the pose cannot be recovered (e.g. second row, last image). However, the reprojection error achieved by our algorithm is clearly low in most cases; the models generally appear in close-to-correct poses. A perfect match between the reprojected model and the observations is not always possible, which is a limitation of the sparse observations and object models we use. Small differences in the reprojection on the image plane may then correspond to large errors

This article is part of Chapter 2: *Pose estimation using a 3D object model* (p.24)
It originally appeared in: *2011 German Conference on Pattern Recognition (DAGM)*

in the actual 3D pose recovered. Most of these errors can be greatly reduced by using additional views of the scene, which is easily done with our method.

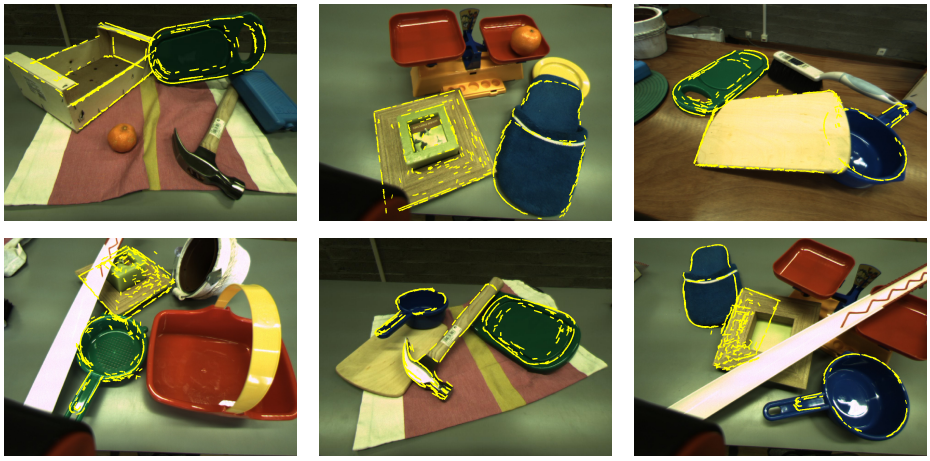


Fig. 5: Results of pose estimation (using a single view), with model features reprojected onto the input image. Most remaining errors are a limitation of the simple object models used, each learned from a single stereo pair.

6 Conclusions

We presented a generic method for 3D pose estimation of objects in 2D images, using a probabilistic scheme for representing object models and observations. This allows the method to handle various types of observations, including features that cannot be matched individually; here we use local edge segments. Using these principles, we showed how to use Metropolis-Hastings MCMC to infer the maximum-likelihood pose of a known object in a novel scene, using a single 2D view of that scene. The probabilistic approach makes the pose estimation process possible without establishing explicit model-to-scene correspondences, as opposed to existing state-of-the-art methods. Together with the use of edge segments as observations, the method allows us to effectively handle non-textured objects. Further, the method extends to the use of multiple views, providing a rigorous framework for integrating evidence from multiple viewpoints of a scene, yielding increased accuracy with only a linear increase of computation time with respect to the number of views. We validated the proposed approach on two publicly-available datasets. One dataset allowed quantitative evaluation; the result of an experiment was compared to the results of an existing method, and showed an advantage in performance for our method. The pose estimation process was also evaluated with success on scenes with clutter and occlusion. Future work will extend the current implementation to the use of other visual features, thereby extending the types of objects that can be handled.

This article is part of Chapter 2: *Pose estimation using a 3D object model* (p.25)
It originally appeared in: *2011 German Conference on Pattern Recognition (DAGM)*

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience. Damien Teney is supported by a research fellowship of the Belgian National Fund for Scientific Research.

References

1. Collet, A., Berenson, D., Srinivasa, S., Ferguson, D.: Object recognition and full pose registration from a single image for robotic manipulation. In: ICRA (2009)
2. Collet, A., Srinivasa, S.S.: Efficient multi-view object recognition and full pose estimation. In: ICRA. pp. 2050–2055 (2010)
3. Detry, R.: A probabilistic framework for 3D visual object representation: Experimental data (2009), `http://intelsig.org/publications/Detry-2009-PAMI/`
4. Detry, R., Piater, J.: Continuous surface-point distributions for 3D object pose estimation and recognition. In: ACCV (2010)
5. Detry, R., Pugeault, N., Piater, J.: A probabilistic framework for 3D visual object representation. IEEE Trans. PAMI 31(10), 1790–1803 (2009)
6. Ekvall, S., Hoffmann, F., Kragic, D.: Object recognition and pose estimation for robotic manipulation using color cooccurrence histograms. In: IROS (2003)
7. Gordon, I., Lowe, D.G.: What and where: 3D object recognition with accurate pose. In: Toward Category-Level Object Recognition. pp. 67–82 (2006)
8. Hsiao, E., Collet, A., Hebert, M.: Making specific features less discriminative to improve point-based 3D object recognition. In: CVPR. pp. 2653–2660 (2010)
9. Klein, G., Drummond, T.: Robust visual tracking for non-instrumented augmented reality. In: ISMAR. pp. 113–122. Tokyo (October 2003)
10. Kraft, D., Krüger, N.: Object sequences (2009), `http://www.mip.sdu.dk/covig/sequences.html`
11. Kragic, D., Miller, A.T., Allen, P.K.: Real-time tracking meets online grasp planning. In: ICRA. pp. 2460–2465 (2001)
12. Krüger, N., Wörgötter, F.: Multi-modal primitives as functional models of hypercolumns and their use for contextual integration. In: Gregorio, M.D., Maio, V.D., Frucci, M., Musio, C. (eds.) BVAI. Lecture Notes in Computer Science, vol. 3704, pp. 157–166. Springer (2005)
13. Mittrapiyanuruk, P., DeSouza, G.N., Kak, A.C.: Calculating the 3D pose of rigid objects using active appearance models. In: ICRA. pp. 5147–5152 (2004)
14. Pless, R.: Using many cameras as one. In: CVPR (2). pp. 587–593 (2003)
15. Pope, A.R., Lowe, D.G.: Probabilistic models of appearance for 3D object recognition (2000)
16. Pugeault, N.: Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation. VDM Verlag Dr. Müller (2008)
17. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. Int. J. Comput. Vision 66(3), 231–259 (2006)
18. Sudderth, E.B.: Graphical models for visual object recognition and tracking. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2006)
19. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3D tracking using online and offline information. IEEE Trans. PAMI 26(10), 1385–1391 (2004)

Chapter 3

Reconstruction of 3D models from multiple views

Following the design of the method of Chapter 2 for pose estimation, which relies on 3D models of objects, we worked on a procedure to actually obtain such models. Even though a number of competing methods simply use perfect, hand-made CAD models, our goal was rather to autonomously learn such models, using observations of the objects of interest obtained during a training phase. The object models initially used in Chapter 2 were obtained with the sparse-stereo system developed by Krüger *et al.*, which reconstructs 3D edges from stereo imagery [42, 43]. Some practical reasons motivated the development of our own procedure. Most importantly, this procedure is not a strictly *stereo* reconstruction method, but it is rather designed to reconstruct models from large numbers of views. This allows producing more complete reconstructions of objects, since these views can cover all sides of the object. Stereo vision, on the contrary, would only reconstruct a model from one viewpoint, the backside of the object being hidden because of self-occlusion.

We implemented our method for reconstructing 3D edges, although its core principles are more generally applicable. As we will describe below, the method is framed in a probabilistic formulation very similar to the one used in the pose estimation method we designed it to work with (Chapter 2).

The typical scenario considered for the autonomous learning of object models involves a robotic agent free to acquire pictures of the object from a number of viewpoints, e.g. with a camera mounted on a robot arm. Inversely, the object could be rotated in a robotic manipulator in front of a fixed camera. In both cases, the input data for the reconstruction is thus a number of registered 2D views. Technically, these 2D observations are then "backprojected" to the 3D world, and we use the intersection of these projections from the multiple views to define a probability density in the 3D world, which corresponds to the reconstruction of the object. We represent this density in a nonparametric manner, with samples drawn from the distribution. Conceptually, this presents clear similitudes with the pose estimation method of Chapter 2. On the one hand, the two methods use the projection of a probability density of 3D observable features into 2D image features, or vice versa. On the other hand, both methods

use a nonparametric representation of densities, for both the 3D object model, and the corresponding 2D observations of this model in the image.

Conceptually, formulating the methods for reconstruction (of 3D models) and pose estimation (using these 3D models) with similar concepts is very satisfactory as they are designed to work together. Practically, this allowed sharing and reusing significant parts of the code. Those seemingly different projects therefore shared a lot of practical developments. The integration of the two methods into a complete practical application is carried out by a Master student of the University of Innsbruck as his Master thesis.

The paper included in the following pages was presented at the 2012 *Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DimPVT)*.

Sampling-based Multiview Reconstruction without Correspondences for 3D Edges

Damien Teney
University of Liège, Belgium
Damien.Teney@ulg.ac.be

Justus Piater
University of Innsbruck, Austria
Justus.Piater@uibk.ac.at

***Abstract*—This paper introduces a novel method for feature-based 3D reconstruction using multiple calibrated 2D views. We use a probabilistic formulation of the problem in the 3D, reconstructed space that allows using features that cannot be matched one-to-one, or which cannot be precisely located, such as points along edges. The reconstructed scene, modelled as a probability distribution in the 3D space, is defined as the intersection of all reconstructions compatible with each available view. We introduce a method based on importance sampling to retrieve individual samples from that distribution, as well as an iterative method to identify contiguous regions of high density. This allows the reconstruction of continuous 3D curves compatible with all the given input views, without establishing specific correspondences and without relying on connectivity in the input images, while accounting for uncertainty in the input observations, due e.g. to noisy images and poorly calibrated cameras. The technical formulation is attractive in its flexibility and genericity. The implemented system, evaluated on several very different publicly-available datasets, shows results competitive with existing methods, effectively dealing with arbitrary numbers of views, wide baselines and imprecise camera calibrations.**

I. Introduction and related work

The problem of 3D scene reconstruction using multiple 2D images from different viewpoints is fundamental in computer vision. The variety of applications, from robotic interaction to phototourism or reverse engineering, has led to the development of numerous methods over the years. These can be broadly classified into two categories: (i) intensity-based multiview stereo methods, which produce *dense* surface reconstructions, and (ii) feature-based methods, which recover *sparse* 3D models of geometric features. Although many of these methods have proven successful in select fields of application, their typical requirements and limitations in operating conditions motivated the development a novel, feature-based method, particularly suited to the use of hard-to-match features. This method, which we successfully applied to the particular problem of 3D curve reconstruction, will be introduced after reviewing related literature.

Methods of the first category mentioned above typically aim at producing detailed 3D reconstructions of objects, enforcing photometric consistency and surface continuity constraints to recover a dense shape description. However, those methods can typically only operate in precisely controlled settings, usually only with Lambertian surfaces, and with large numbers of precisely calibrated cameras. Those typical requirements for controlled acquisition conditions often prove impractical for general applications (see [1] for a review). While dense reconstructions can offer visually striking results, there are many applications where sparse reconstructions are sufficient, as argued below.

Methods of the second category aim at reconstructing sparse 3D models, made up of isolated geometric features, such as points or edges. Such methods are particularly interesting as they provide more expressive and efficient representations than dense surfaces, typically at a fraction of the computational cost. The classical methods rely on the detection of interest points in the individual 2D views, and then use their local appearance (e.g. using SIFT descriptors [2]) to propose likely matches between observations from different views. The geometric consistency between pairs or triples of points can then be enforced using the well-known epipolar or trifocal constraints [3], effectively leading to the reconstruction of a 3D point cloud compatible with the observations. The first limitations of this approach are obviously those of the extraction and matching of image features, which works best on texture-rich images, but can perform poorly on scenes with mostly homogeneous surfaces or little detail [4]. Moreover, the matching of local appearance descriptors is made harder as the baseline between the considered viewpoints increases [5], practically limiting this approach to the consideration of close pairs of views at a time.

Other methods of the second category make use of image curves, or edges, extracted in the available 2D views [4], [6]–[9]. Reconstructions made up of edge segments convey more geometric information than point clouds [6] and offer greater invariance to changes in illumination and viewpoint. Edge-based reconstructions have moreover proved directly useful for practical applications like pose estimation [10], [11], or the prediction of grasping points of objects [12]. The classical approach, described above, of matching observations between different views (now lines or cuves) is however a non-trivial problem [6], exacerbated by the variability in the extraction of said edges from the 2D images. Li *et al.* [4] reviewed various schemes, e.g. using extended projective

This article is part of Chapter 3: *Reconstruction of 3D models from multiple views* (p.29)
It originally appeared in: *2012 Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DimPVT)*

geometry [13] or differential geometry [4], or restricting the problem to closed curves [14]. Common drawbacks are strong requirements for precisely calibrated camera [4], [9], [13] and limitations to pairs or triples of views at a time [13]. In [15], Kaess *et al.*focuses on the subproblem of fitting parametric curves to contours identified in several images, using a Monte Carlo-type search as we do. They do not however consider the reconstruction of entire scenes with several objects and the inevitable uncertainty in the input observations. Kahl *et al.* [7] present an approach that also avoids establishing correspondences between views, but delivers results only on simple scenes, reconstructing only small numbers of short curve fragments. We present results on arguably more challenging datasets and in much more varied conditions (see Section IV).

Multiview reconstruction is part of the larger problem of simultaneous localization and mapping (SLAM). In contrast to SLAM, this paper assumes calibrated views and does not make use of core assumptions made by most SLAM methods, most importantly the abundance of input views and feature tracking across views. Some SLAM methods are nevertheless relevant to the current discussion. Klein *et al.* [16] use edges as image features and show how complementary they are to interest points. They focus on the localization problem, and do not deliver convincing results for reconstruction of said edges. Civera *et al.* [17] propose, as we will do, an alternative probabilistic formulation to the classical Gaussian measurement uncertainty, but also focus on localization. [18] goes beyond precisely localizable features by tracking surface patches under photometric constraints to provide a dense reconstruction, but is based on frame-to-frame tracking.

The method proposed in this paper aims at reconstructing a sparse 3D model of geometric features. The key principle is the definition, using each available 2D view, of a probability density in the 3D reconstructed space, which is *compatible* with the view considered. This distribution thus encompasses all backprojected 3D features that could have produced the considered image. Considering all available views, the intersection, or product, of those distributions is then proposed as the distribution of 3D features of the reconstructed scene. We present in Section II an efficient algorithm for obtaining individual samples from that distribution, effectively yielding a set of 3D features (edge fragments in our implementation) describing the reconstructed scene. A second algorithm is proposed that iteratively identifies contiguous regions of high density in the 3D space, which links such samples together, forming continuous 3D curves.

The strength of the proposed approach is to handle non-precisely localizable features, which cannot be matched one-to-one, or which present uncertainty in some dimension of the observation (like a point along an edge). The resulting curve reconstruction method therefore does not rely on connectivity in input images, effectively accounting for the variability in the extraction of edges from the images. Other reconstruction methods have been designed to handle uncertainty in the input data, often by relaxing the matching and geometry constraints. For example, Fabbri *et al.* [6] implemented a two stage process, where an initial robust reconstruction is used to optimize the calibration of the cameras, to then obtain a finer reconstruction in the second stage. That approach, which can be traced back to the classical RANSAC algorithm, proved robust, but, in addition to being arguably computationally inefficient, lacks the genericity and flexibility of the formulation presented below. Note finally that similar probabilistic models of objects and image observations have been used in the past [10], [11], and this work can be seen as their extension to the problem of 3D reconstruction.

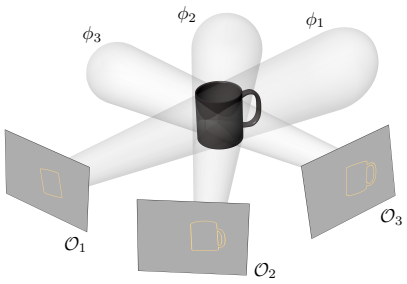


Figure 1. The proposed method uses the observations of each input view, $\mathcal{O}_k$, to define probability distributions ϕ_k in the 3D, reconstructed space; the reconstructed model lies at the intersection of those distributions.

We must finally remark that reconstruction without correspondences is not new. A basic formulation of the problem was presented in [19]. In [20], Dellaert *et al.*used expectation-maximation to recover the structure of a scene, handling however only precisely localized features, and only presented results on toy examples under several unrealistic assumptions. More recently, [21] showed how to recover the camera transformation between pairs of views using the radon transform, but without considering the 3D structure of the scene at all.

II. Probabilistic reconstruction from 2D views

We now present the proposed method, first in a general formulation, then applied to the use of edge segments. Those features correspond, in the input images, to points extracted along lines of maximum gradient, and characterized by their position and orientation on the image plane (see Section III-A). In the reconstructed model, they correspond to oriented 3D points, that we typically represent by short, fixed length, 3D line segments (see Fig. 4b for example); they can be connected together to form continuous curves (e.g. 4c).

A. Probability distributions from image observations

The key idea of the method is to define, from each available 2D view, a probability density over the reconstructed 3D

space, which is *compatible* with the observations in that view (1). In other words, it describes the distribution of backprojected 3D features that could have produced the considered image, given the uncertainty present in that image, and in the available estimation of the camera parameters. Formally, each view $k \in [1, N]$ is described by a set of image features, or *observations*

$$\mathcal{O}_k \;=\; \{y_i\}_{\,i \,\in\, [1,M_k]} \;, \tag{1}$$

where $y_i \in \mathbb{R}^2 \times \mathcal{A}$ are the image features, characterized by their position in the image, and some descriptor in an appearance space $\mathcal{A}$. In the case of edge segments, which have an orientation but no direction, the appearance descriptor is an element on the semicircle (i.e. an angle in $[0, \pi[$), and $\mathcal{A} = S_1^+$. Considering instead more classical interest points, described by their position in the image and their local appearance, the space $\mathcal{A}$ would then contain normalized texture descriptors. The 3D, reconstructed model, is to be defined on a corresponding space $\mathbb{R}^3 \times \mathcal{A}'$. With edge segments, then characterized by a 3D orientation, we have $\mathcal{A}' = S_2^+$.

We will now define, for a view k, a probability distribution ϕ_k on the reconstructed space, using kernel density estimation (KDE). Each element y_i of the considered view is associated with an element of the reconstructed 3D space, $y'_i \in \mathbb{R}^3 \times \mathcal{A}'$. This element can simply be obtained by setting a normalized value for the extra dimensions; e.g., the depth and 3D orientation of our edge segments can be fixed to lie on the image plane in the 3D world (see Fig. 2). This now allows us, using KDE, to define the distribution ϕ_k by its probability density function

$$\phi_k(x) \;=\; \frac{1}{M_k} \sum_{i=1}^{M_k} K_i(y'_i, x) \;, \tag{2}$$

where K_i are kernel functions on $\mathbb{R}^3 \times \mathcal{A}'$. Intuitively, one kernel $K_i(y'_i, x)$ models the distribution of all reconstructed features that could have produced the observation y_i. The details, which will depend on the type of features used, are straightforward in the case of edge segments. Looking at the position only, it represents a constant probability density along the backprojected ray (see Fig. 2). Formally, we measure the distance between a given y'_i and x by 3 scalars:

i. d_1, the closest distance in position between x and the line defined by y'_i (the backprojected ray),
ii. d_2, the depth of x, relative to the camera center,
iii. d_3, the difference in orientation between x, and the plane corresponding to the backprojection of the orientation of y'_i.

We then define our kernel function $K_i(y'_i, x)$ as the product of 3 independent kernels that make use of those distance measures: a Gaussian kernel on (d_1/d_2) (inducing a conical surface of equidensity for the position, see Fig. 2), a box kernel on d_2, and a von Mises-Fisher kernel on d_3 (which is a Gaussian-like distribution on orientations [22]). Note that the effect of the box kernel on the depth only corresponds to fixing a hard threshold on the distance to the reconstruction. Indeed, the only assumption that can generally be made here is that a reconstructed point must lie in front of the camera, and within a realistic depth range.

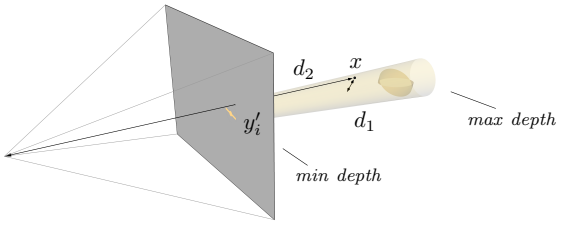


Figure 2. Illustration of an observation y'_i (an oriented point on the image plane) and its associated kernel $K_i(y'_i, \cdot)$ in the reconstructed, 3D space, both for the position and the orientation (*surfaces of equidensity in transparent orange*). The kernel, evaluated at a point x, uses the distances d_1 and d_2, resp. to the axis of the backprojected cone and to the camera center (see text for details); d_3 is not represented.

The geometric meaning of our definition of a kernel is quite intuitive, and is illustrated in Fig. 2. For example, the surfaces of equidensity for the position in the 3D space correspond to truncated cones, extending along the camera's projection rays. The selection of the bandwidth of the kernels is discussed in Section III-B.

The definition of the kernels could be extended to other types of image features, or to include edge curvature for example. We propose another minor extension that takes into account the uncertainty along the orientation of an edge, thereby "flattening" the cone of Fig. 2. For this purpose, the distance d_1 is separated in 2 components d'_1 and d''_1, respectively aligned and orthogonal to the orientation of the edge; they are then simply evaluated as (d'_1/d_2) and (d''_1/d_2) in Gaussian kernels of respectively large and small variance, thus allowing more slack along the orientation of the edge (see specific results in Section IV-A).

Finally, as a side note, let us remark that defining a probability distribution over the *reconstructed* space, as we did, differs from the classical formulation of the problem, where the reconstructed model is compared, once reprojected in the image space, against the 2D input observations. We will remark that, under certain parameterizations, the two approaches can be rendered equivalent. Our formulation was however chosen in this presentation, as it offers a more intuitive formulation of the sampling-based reconstruction methods that we will propose below.

B. 3D Reconstruction of individual points

The probability distributions ϕ_k we have defined make each use of one single view. We now combine them to produce another distribution ψ in the reconstructed space

that is globally consistent with all available views. It is given by its probability density function

$$\psi(x) \;=\; \frac{1}{C} \prod_{k=1}^{N} \big(\phi_k(x) + \varepsilon\big) \;, \tag{3}$$

where C is a normalization constant, and ε is a fudge constant, small relative to the scale of $\phi_k(x)$. This definition practically uses the intersection of the ϕ_k, relaxed by the constant ε. This allows observations that appear in some but not *all* input images to produce a nonzero density region in the reconstructed space. This proves necessary in practice, to handle e.g. self-occlusions and missing observations.

Equation (3) gives a formal definition of the 3D reconstruction of the scene. The main goal however is to obtain an explicit and practical representation of this model. Sampling directly from ψ is generally not feasible, but we propose an approximate method based on importance sampling (see for example [10], [23]). Importance sampling (IS) allows one to sample a target distribution $p(x)$, assuming one can evaluate $p(x) = \bar{p}(x)/Z$ up to some normalization constant Z, by using samples x^ℓ from a *proposal distribution* p', ideally similar to p. IS accounts for the difference between the target and proposal distributions by assigning to each sample x^ℓ a weight given by

$$w^\ell \;=\; \bar{p}(x^\ell) \;/\; p'(x^\ell) \;. \tag{4}$$

The collection of weighted samples $\big\{(x^\ell, w^\ell)\big\}_{\ell=1}^{L}$ is then, under mild assumptions, asymptotically consistent with the target distribution. This procedure is obviously most efficient as the proposal distribution is close to the target distribution. In practice, the collection of weighted samples is then generally resampled, to a smaller set of L' ($< L$) *unweighted* samples.

The proposal function used here is given by

$$\psi'(x) \;=\; \frac{1}{C'} \sum_{\substack{(k_1,k_2) \\ \in \text{ pairs}(1,N)}} \phi'_{k_1}(x)\; \phi'_{k_2}(x) \;, \tag{5}$$

where C' is a normalization constant, and $\text{pairs}(1, N)$ denotes the list of all unique pairs of indices between 1 and N. Each density function ϕ'_k is a variation of the ϕ_k defined above, in which the kernels used are all box functions. Intuitively, ψ' simply corresponds to all the intersections of pairs of views. Sampling from $\psi'(x)$ is easily done by choosing two arbitrary views k_1 and k_2, and triangulating two random observations y_1 and y_2 from each, the kernels of which intersect at least by a small amount (i.e. the 3D projections of which intersect each other within a small threshold). The bandwidth of the box kernels of ϕ' will be chosen so that they extend up to a reasonable cutoff threshold of the exact kernels of ϕ. This ensures that the proposal distribution ψ' will generate samples in all of the most interesting regions of the target distribution ψ. The weights assigned to the proposal samples of ψ' are then simply computed using (4). They can then be resampled to obtain a set of non-weighted points.

C. 3D Reconstruction of continuous curves

The method presented above reconstructs individual points as samples from a probability distribution in the 3D space. Some interesting parts of the scene may however correspond to regions of lower density (e.g. due to missing observations in one or several views), but which can however still be identified as local maxima. Moreover, in the particular case of curve reconstruction, one wants to reconstruct continuous curves, and not individual points. Those two objectives can be met through the iterative procedure described below, which uses the individual samples as starting points for a stochastic exploration of the reconstructed space.

For each reconstructed curve, the procedure starts with a sample $x_0 \in \mathbb{R}^3 \times S_2^+$. It then iterates, searching at each step for a point x_{i+1} along a ridge of locally maximum probability density. Formally, local proposals are generated from a point $x = (p, \theta)$ of position $p \in \mathbb{R}^3$ and orientation $\theta \in S_2^+$ (a unit 3-vector), as a set of L samples:

$$\textit{proposals}(x) \;=\; \big\{\big(\, p + \Theta_\kappa(\theta) * \Gamma_{(\alpha,\beta)}, \Theta_\kappa(\theta)\,\big)_j\big\}_{\,j \,\in\, [1,L]} \;, \tag{6}$$

where Γ is a gamma distribution that generates the distance in position to a proposal, and Θ is a Von Mises-Fisher distribution used to randomize the orientation. This uses the assumption that the next point of the curve is most likely in the direction of the current point. The parameters κ, α, β define how "spread out" the proposals are from an exactly straight line. The likelihood of each proposal is evaluated (Eq. 3), and the best one is selected as the new point x_{i+1} of the curve. The procedure is repeated, unless the likelihood of all L proposals fall below a threshold, indicating the probable end of the curve. That threshold is fixed beforehand as fraction of the mean density of a batch of samples of the whole scene. The procedure is comparable to the classical Canny algorithm, which, likewise, follows ridges of local optima until falling below a predefined threshold. Note that the use of a purely random walk scheme for selecting the neighbours in our method — as opposed to estimating local derivatives of a likelihood function (as could be done using differential geometry) — is motivated by the genericity of the procedure, which we plan to apply to other types of image features as future work. Finally, as the scene is being reconstructed, we "prune" ψ, removing the kernels that have significant overlap with the curves already reconstructed. This helps reconstructing parts of the scene of low probability density, initially masked out by regions of higher density, and also avoids reconstructing several times the same portions of a scene.

III. Implementation

A. Edge detection in input images

The image features we use are oriented 2D points, identified along the edges in the images. We selected the method of [24], which is a simple method based on image gradients that extracts the orientation of the edges significantly better than the traditional method, which simply uses the direction orthogonal to the gradient. That method was chosen instead of more sophisticated ones which take texture or global segmentation into account, as they can be extremely slow and are thus not an option for many applications of 3D reconstruction. This also ensured a fair comparison with other published methods which used basic gradient-based edges as well.

B. Choice of parameters of the reconstruction

The kernels associated with the observations are parametrized by their bandwidth in position and orientation. This size should reflect the estimated uncertainty in the input data, and can be set according to a small fraction to the estimated scale of the scene. Our experiments showed however that the method was not particularly sensitive to the choice of those parameters. For example, in the experiments (with both small and large camera calibration errors) of Section IV-D, with 640×480 pixel images, the size of the kernels was set to allow a corresponding maximum deviation in the images of about 12 pixels and $20°$.

The parameters used for local proposals (Eq. 6) are also to be set relatively to the scale of the scene. For example, the scene of Section IV-D, measuring about 1000 mm in diameter, used local proposals corresponding to a spacing of 5 mm and a deviation in orientation of $15°$ on average, with $L = 50$.

Finally, the running time of the iterative procedure grows linearly with the number of reconstructed points. The cost associated with the reconstruction of a point mostly corresponds to the evaluation of ψ (Equation 3) for the proposals. One evaluation involves the processing of every kernel of every input view, and is thus $O(N\bar{M})$, where N is the number of views, and $\bar{M}$ the average number of observations per view. We currently use this basic implementation. However, a cleverer implementation could efficiently preselect the few kernels likely to be relevant to the evaluation of a given point, using an ordered data structure. Since the influence of a kernel in its distribution drops below insignificant values past some distance, one could, in this way, restrict the evaluation of the kernels to a small fraction of them.

IV. Experimental results

The proposed method was evaluated on 4 very different datasets. It is notoriously hard to produce ground truth reconstructions for evaluating feature-based methods, due to the ambiguous selection of the features to reconstruct. Datasets for benchmarking dense reconstruction methods have been produced; however, the ground truth model is not necessarily made public [1], and the selection of actual edges from continuous surfaces [25] or 2.5D models makes it hard to design a meaningful quantitative evaluation of a method like ours. Competing methods for curve reconstruction faced a similar situation, which explains why no extensive qualitative evaluations were published. [6] made an exception, but they only evalute their ability to match correct curve fragments between views, using a set of manually labelled ground truth correspondences — which was unfortunately not made public.

Practically, our prototype software was implemented in Matlab. Running times of such an implementation (especially of an iterative method) have little meaning, as the only switch to a compiled language offers potentially enormous room for improvement. Bearing this in mind, we report, as a base point, that a reconstruction as shown in Fig. 4 or Fig. 6a currently takes about 2 to 5 minutes on a standard laptop without multithreading. Most recent competing methods do not discuss the issue of efficiency; Fabbri *et al.* [6] report running times in the order of minutes on scenes like the dinosaur (see below). Let us note moreover that most parts of our algorithm are straightforward to parallelize.

A. Synthetic toy example

The lack of datasets with proper ground truth motivated the use of a synthetic toy example, in order to evaluate and demonstrate basic properties of the proposed method. The scene, pictured in Fig. 3a, contains curves of various lengths and shapes. Their exact 3D shape is used to directly generate the 2D edge maps used as input to the reconstruction method. This bypasses the stage of edge extraction from 2D images, focusing this evaluation on the reconstruction process alone. To simulate realistic conditions and missing observations, random parts of the curves are masked when generating those edge maps. The scene itself measures about 500 mm in diameter; we use 7 views from different viewpoints around the scene, at a distance of approximately 900 mm.

We compare reconstructions and ground truth using the accuracy/completeness metrics proposed in [1]. To obtain accuracy, we measure the Euclidean distance from each reconstructed point to the closest ground truth curve. The accuracy is then defined as the distance so that 90% of the points fall below that threshold. To obtain the completeness, we consider a number of points sampled uniformly along the ground truth curves, and count the ratio of them that have a part of the reconstruction within a reasonable distance ($15°$ in orientation, and $5/8$ mm in position for scenes without/with noise). The exact choice of those thresholds is not relevant here, since we use them to compare different methods and not to obtain absolute performance values. We report accuracy/completeness scores for 4 different reconstruction methods: (i) a baseline method where we perform

random triangulations (Eq. 5), and keep a fixed number (1000) of points with a probability density (computed as in Eq. 3, but without orientation) above a threshold; this corresponds approximately to the basic approach where one simply imposes a maximum 2D distance between the re-projected reconstruction and the input observations; (ii) our sampling method (Section II-B) used to recover the same number (1000) of points; (iii) our iterative method for curve reconstruction (Section II-C); (iv) the same method accounting for uncertainty specifically along the orientation of edges (Section II-A).

Each method is run with different lower thresholds on the probability density of reconstructed points, setting the tradeoff to be made between accuracy and completeness. We report results in Fig. 3b, with and without noise on camera calibrations (in the form of added Gaussian perturbations of $\sigma = 4\,\text{mm}$ on the camera positions). We also plot, in Fig. 3c, the accuracy and the local probability density (Eq. 3) of a number of random samples (Eq. 5). This allows verifying that there is indeed a correlation between the probability density obtained through our definition, and the actual correctness of a reconstructed point. Reconstructions showing good accuracy can however sometimes correspond to low probability densities, which explains why our sampling method alone cannot recover the entire scenes, as opposed to the iterative method. Moreover, we also verify that the correlation between accuracy and probability density still holds when adding noise (as above) to the camera calibrations.

B. Dinosaur

The "dinosaur" dataset is standard for the evaluation of dense reconstruction methods [1]; we use the version made of 16 views from a circle around the object. We show in Fig. 4b a reconstruction made of individual points, obtained using our sampling method. These samples are drawn mostly in the regions of high probability density of the reconstructed space, with more samples in the regions the most precisely defined, e.g. along the crest on the back of the animal (such sharp edges correspond to well-defined edges in the 2D images). In Fig. IV-Bc, we show a reconstruction of continuous curves of the same scene; those curves are correctly identified along ridges of local maxima of the probability density function, yielding a high quality reconstruction of the object. Those results, directly comparable with those presented in [6], show a clear advantage, particularly in the level of noise in the reconstruction.

C. High-resolution building

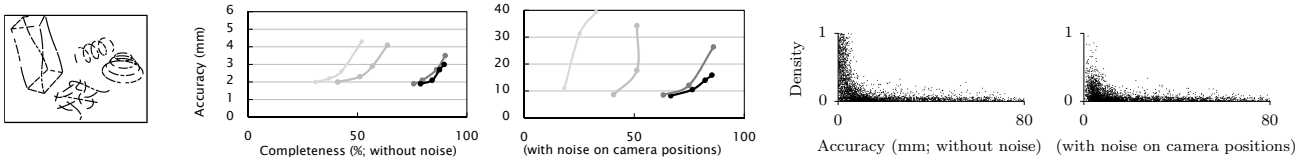
Strecha *et al.* [25] produced a dataset of high resolution pictures of buildings for evaluating dense reconstruction methods. We chose to evaluate our method on one of those scenes ("Herz-Jesu-P8") as it represents a very different type of input data than our other evaluations. The images are of high resolution, but the nature of the scene (very textured surfaces and lots of fine details) renders the extraction of stable edges from the 2D images a difficult problem already. The reconstructed 3D model (Fig. 5a-b) exhibits missing parts, which are a direct consequence of this problem (corresponding to missing observations in the input data). The 8 viewpoints span only a small arc, roughly in the same plane, leaving a great deal of uncertainty in the *depth* dimension, in particular for the edges parallel to that plane. This can be observed when viewing the reconstructed model from the top (Fig. 5b), as some supposedly straight edges meander in this dimension. The same curves however, when reprojected on an input image, always closely match the input images (Fig. 5c). [25] uses a particular distance measure to evaluate dense reconstruction methods, requiring non-public information (calibration uncertainty), which prevented direct performance comparisons.

D. Office desk

We finally consider an indoor scene, containing typical household items with little texture (see Fig. 6a), shot from 12 different viewpoints around them. This represents the type of scenes that motivated our approach, in the context of robotic applications, where a robot would take the pictures using an arm-mounted camera. The extrinsic calibration of the camera would thus be known with a precision corresponding to the accuracy of the robotic arm. In this evaluation however, and for purely practical reasons, we used a checkerboard pattern in the scene with standard calibration software. We obtained visually excellent reconstructions (Fig. 6). A challenging part of the scene is the checkerboard, as it contains many lines close together, in both similar and different orientations. We then intentionally added noise to the positions of the 12 cameras, to verify the influence on the reconstruction (see Fig. 6b-e for details). The highest tested level of additional noise, drawn from a Gaussian distribution of $\sigma = 8\,\text{mm}$, introduces corresponding translation errors as large as 10 pixels on the 640×480 pixel images. Experiments show that a reconstruction is still possible; some regions of the reconstructed space now receive a probability density lower than the allowed threshold, explaining missing parts in the reconstruction. Those results are representative of the performance we obtained on many experiments of similar nature.

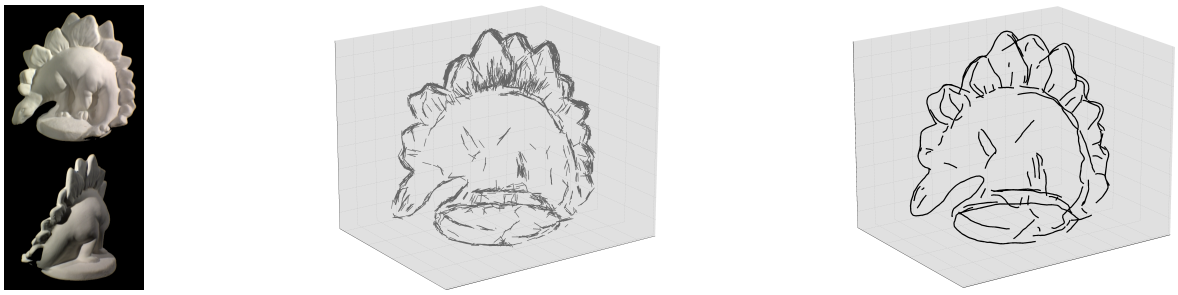
V. Conclusions and Future Work

We presented a novel method for feature-based 3D reconstruction from multiple calibrated views. We introduced a probabilistic formulation that admits hard-to-match features particularly suited to edge segments. The reconstructed scene is modelled as a probability density in the 3D space, from which we can draw individual samples. Those are then used as starting points to reconstruct continuous 3D curves. The effectiveness of the approach was demonstrated



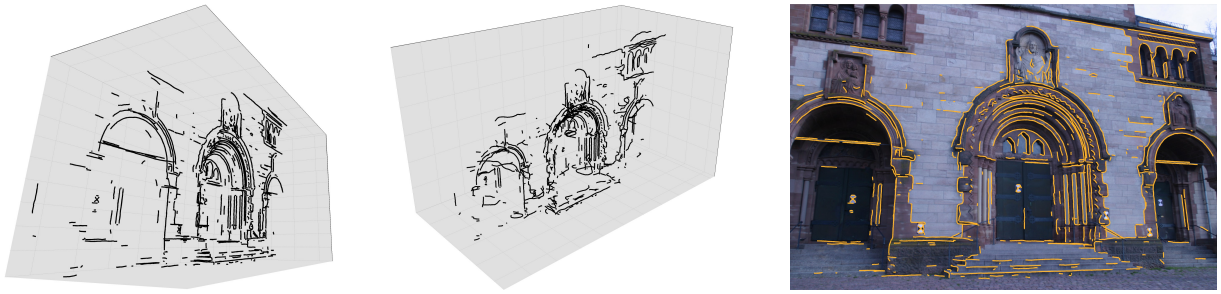
(a) (b) (c)

Figure 3. Evaluation on synthetic scene: (a) example input edge map, note missing observations; (b) accuracy/completeness scores for (light to dark) random sampling based on position only, our sampling method using orientation, our iterative method with conical kernels, then with "flattened" kernels (accounting for uncertainty along the edge orientations); (c) density/accuracy of random samples (density evaluated up to a multiplicative constant).



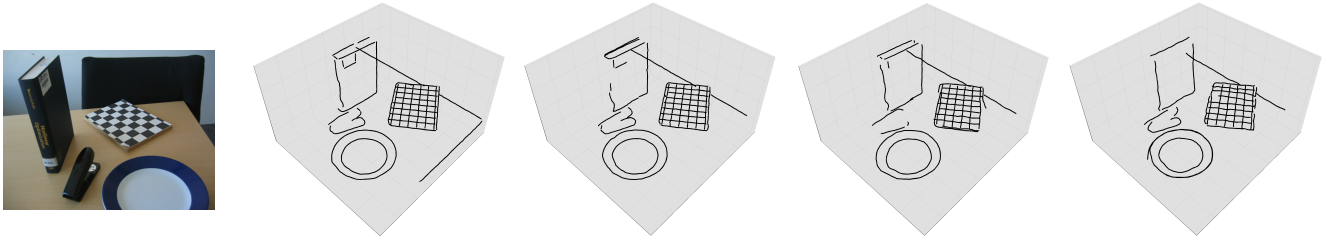
(a) (b) (c)

Figure 4. (a) Example images of the dinosaur dataset; (b) individual reconstructed 3D points obtained through our sampling method; (c) reconstruction of continuous curves.



(a) (b) (c)

Figure 5. (a) Reconstruction of the building dataset, missing parts are mostly due to missing observations, difficult to extract from the input images; (b) other view of the reconstruction, showing the imprecisions in depth, as the input viewpoints span only a small arc in front of the building; (c) reconstructed edges, reprojected on an input image, match however closely.



(a) (b) $\sigma = 0\,\text{mm}$ (c) $\sigma = 2\,\text{mm}$ (d) $\sigma = 4\,\text{mm}$ (e) $\sigma = 8\,\text{mm}$

Figure 6. Reconstruction of scene with error in camera calibration; one input image (a); 3D reconstructions (rendered from a novel viewpoint) with original estimated camera calibration (b) and with added perturbation on camera position from Gaussian noise of variance σ (c-e); significant levels of error still allow reconstruction, at the price of some imprecisions (plate, checker board) and missing edges (book, lower edge of the table).

This article is part of Chapter 3: *Reconstruction of 3D models from multiple views* (p.35)
It originally appeared in: *2012 Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DimPVT)*

on existing and new datasets, and showed competitive results with an existing method, while exhibiting more technical flexibility and genericity in its formulation. An important direction for future work is the evaluation of this method on features other than edges.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience. Damien Teney is supported by a research fellowship of the Belgian National Fund for Scientific Research.

References

[1] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 519 – 528. 1, 5, 6

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints." *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. 1

[3] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004. 1

[4] G. Li, Y. Gene, and S. Zucker, "Multi-view edge-based stereo by incorporating spatial coherence," in *3-D Digital Imaging and Modeling, 2007, Sixth International Conference on*, 2007, pp. 341 –348. 1, 2

[5] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3d objects," *International Journal of Computer Vision*, vol. 73, pp. 263–284, 2007. 1

[6] R. Fabbri and B. Kimia, "3D curve sketch: Flexible curve-based stereo reconstruction and calibration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1538 –1545. 1, 2, 5, 6

[7] F. Kahl and J. August, "Multiview reconstruction of space curves," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 1017 –1024 vol.2. 1, 2

[8] S. Liu, K. Kang, J.-P. Tarel, and D. B. Cooper, "Free-form object reconstruction from silhouettes, occluding edges and texture edges: A unified and robust operator based on duality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 131–146, 2008. 1

[9] K. Potsch and A. Pinz, "3D geometric shape modeling by '3D contour cloud' reconstruction from stereo videos," in *Computer Vision Winter Workshop 2001*, Mitterberg, Austria, 2001. 1, 2

[10] R. Detry, N. Pugeault, and J. Piater, "A probabilistic framework for 3D visual object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1790–1803, 2009. 1, 2, 4

[11] D. Teney and J. Piater, "Probabilistic Object Models for Pose Estimation in 2D Images," in *DAGM*, ser. LNCS, vol. 6835/2011. Heidelberg: Springer, 2011, pp. 336–345. 1, 2

[12] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger, "A strategy for grasping unknown objects based on co-planarity and colour information," *Robotics and Autonomous Systems*, 2010. 1

[13] C. Schmid and A. Zisserman, "The geometry and matching of lines and curves over multiple views," *International Journal of Computer Vision*, vol. 40, no. 3, pp. 199–233, 2000. 2

[14] R. Berthilsson, K. Astrom, and A. Heyden, "Reconstruction of general curves, using factorization and bundle adjustment," *International Journal of Computer Vision*, vol. 41, pp. 171–182, 2001. 2

[15] M. Kaess, R. Zboinski, and F. Dellaert, "Mcmc-based multi-view reconstruction of piecewise smooth subdivision curves with a variable number of control points," in *European Conference on Computer Vision (ECCV)*. Springer, 2004, pp. 329–341. 2

[16] G. Klein and D. Murray, "Improving the agility of keyframe-based slam," in *European Conference on Computer Vision (ECCV)*, 2008. 2

[17] J. Civera, A. J. Davison, and J. M. M. Montiel, "Unified inverse depth parametrization for monocular slam," in *Robotics: Science and Systems*, 2006. 2

[18] R. Newcombe, S. Lovegrove, and A. Davison, "DTAM: Dense Tracking and Mapping in Real-Time," in *IEEE International Conference on Computer Vision (ICCV)*, 2011. 2

[19] Y. qing Cheng, R. T. Collins, A. R. Hanson, and E. M. Riseman, "Triangulation without correspondences," in *DARPA Image Understanding Workshop*, 1994, pp. 993–1000. 2

[20] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun, "Structure from motion without correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000. 2

[21] A. Makadia, C. Geyer, and K. Daniilidis, "Correspondence-free structure from motion," *International Journal of Computer Vision*, vol. 75, no. 3, pp. 311–327, 2007. 2

[22] R. A. Fisher, "Dispersion on a sphere," in *Proc. Roy. Soc. London Ser. A.*, 1953. 3

[23] E. B. Sudderth, "Graphical models for visual object recognition and tracking," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2006. 4

[24] A. Tamrakar and B. B. Kimia, "No grouping left behind: From edges to curve fragments." in *IEEE International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8. 5

[25] C. Strecha, W. von Hansen, L. J. V. Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 5, 6

Chapter 4

Recognition using 2D training examples

The remaining work presented within this thesis, starting with this chapter, is concerned with the recognition and pose estimation of objects in 2D images, directly using 2D example images as training data, and thus without any explicit 3D reconstructions. The motivations for so-called "exemplar-based" methods are multiple. First, removing the need for a 3D model may avoid the potentially computationally costly reconstruction step. Second, although this is still debated, motivation for exemplar-based approaches has been suggested by the human visual system, which was shown to exhibit properties of a view-based lookup function when recognizing objects, being robust to small changes of about 20° around trained viewpoints [51]. Third, such methods do not require all the training examples to be initially available, and these examples can be added incrementally to the model. Fourth, 2D example images can be argued to better represent and encode the appearance of objects than 3D models, which are mainly concerned with the geometric aspects of the shape of the object.

The proposed system tackles the task of detection in clutter at the same time as pose estimation, as demonstrated through our evaluation on images presenting large numbers of objects stacked onto each other. In order to perform pose estimation with exemplar-based methods, the basic idea is to identify, in the test image, one of the provided training examples. Continuous pose estimation, which goes beyond this mere classification among learned viewpoints, is only possible through more complex procedures. Ours relies on a probabilistic voting in the 6-DoF pose space. The whole method can actually be compared to a generalized Hough transform at a very large scale, i.e. with votes made of combinations of image features from the test view, and image features of all training views at the same time. These votes are then cast in the 6-DoF pose space, accounting for localization, scale and rotation in the image, but also for the viewpoint. Details on the implementation of this procedure had to be kept short in the paper included below, due to imposed maximum length. The technical complexity may thus not be apparent, but the implementation required particular care in order to make computations practically tractable. After computing the actual votes in the pose space, the challenge is to identify the regions of high density in this high

dimensional space, typically from millions of votes. We use a two-step procedure to do so. First, a histogram approximates the density in regular bins and serves to eliminate those of low density. In the small number of remaining regions, we then perform a proper kernel-based evaluation of the density, to finally identify local maxima which give us candidate detections of the object in the image, together with their corresponding full pose. The effort, in terms of software engineering, necessary to make these computations tractable lies mostly in the usage of the memory for storing the votes, and for representing the bins of the histogram trough efficient data structures.

The paper also includes a way of fusing results from multiple sources of evidence, basically from multiple runs of the above procedure. These different runs could use different images of a same scene, or different types of image features. We obtain an "average" of the results of all runs through a probabilistic voting in the pose space. This proved practically very interesting, as demonstrated by some of our experiments. These additional results are however clearly anecdotal compared to the main contributions of the paper. They also did not necessitate much additional effort to produce, as they share the concepts and parts of the code (for voting and averaging in the pose space) of the main procedure.

4.1 Full algorithm for pose estimation

We give in Algorithm 1 the explicit listing of the procedure for pose estimation introduced above. It should be read together with Section III of the paper included in the following pages.

The running time of the algorithm is dominated by the execution of its first (**for**) loop. The innermost operations of this loop process every pose compatible with every sample x_i obtained from the observations. The minimal number of necessary samples to use (N_{samples}) is hard to determine, and depends heavily on the amount of clutter in the test image. In general, it will therefore be chosen as a fraction of the total number of observations (N). The number of poses that we can identify as compatible with each of these samples is then proportional to the number of training points (M). As a consequence, this loop, and the whole algorithm, are $O(MN)$. Note that this costly implementation, which makes use of all training pairs for each sample from the test view, could potentially be replaced by a stochastic use of the probabilistic representation of the training data ϕ, using a limited number of random samples thereof at each iteration. This scheme was used in the related use of probabilistic 3D models, by Detry *et al.* [41] and in Chapter 2, but was not investigated here. It is however an obvious — and simple — improvement to reduce computational requirements. It will be used in the improved versions of the method presented in the next chapters.

4.2 Histogram on the 6-DoF pose space

The algorithm presented in Section 4.1 relies on a histogram-based method to identify the regions of the pose space of significant density. To ensure consistent results, it is

Algorithm 1 Algorithm for pose estimation by exhaustive search for local maxima in the pose space.

Input: training pairs $\mathcal{T} = \{(w_i, x_i)\}_i$
distribution of observations ϕ
Output: set of poses $\mathcal{R} = \{w_{*_i}\}_i$

Procedure:
Initialize each bin b of histogram on SE(3): b.contents $\leftarrow \emptyset$
$\mathcal{T}' \leftarrow \emptyset, \quad \mathcal{T}'' \leftarrow \emptyset, \quad \mathcal{R} \leftarrow \emptyset$
for $i = 1..N_{\text{samples}}$ *fill histogram*
  Get sample $x_i \sim \phi$
  foreach training pair $(w, x) \in \mathcal{T}$
    foreach $p \in \mathcal{P}$ such that $f\big((w, x),\ p\big) = (w', x_i)$
      $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{(w', x_i)\}$
      $b \leftarrow \text{findBin}(w')$
      b.contents $\leftarrow b$.contents $\cup \{w'\}$
foreach bin b *select histogram bins*
  if number of elements in b.contents is significant
    $\mathcal{R} \leftarrow \mathcal{R} \cup b$.contents
    $\mathcal{T}'' \leftarrow \mathcal{T}'' \cup \{(w, x) \in \mathcal{T}' : \text{findBin}(w) = b\}$
foreach $w_* \in \mathcal{R}$ *use pre-selected data for kernel-based evaluation*
  Evaluate $p(w_*)$ using kernels supported by elements of $\mathcal{T}''$ only
  if $p(w_*) < \text{threshold}$
    $\mathcal{R} \leftarrow \mathcal{R} \setminus w_*$

necessary to establish a uniform partitioning of the 3D pose space, $\mathrm{SE}(3)$, which is not a trivial problem. Ideally, each bin of the histogram must present a similar area, as well as a similar topological shape. Using the factorization $\mathrm{SE}(3) = \mathbb{R}^3 \times SO(3)$, these requirements can be easily satisfied for $\mathbb{R}^3$ with a uniform division along each of its dimensions. Moreover, in practice, we restrict ourselves to a bounded subspace of $\mathbb{R}^3$. This is not a problematic restriction, since the pose of the object can generally be assumed to lie in a frustum defined by the border of the input image and a reasonable maximum depth. The complex topology of $SO(3)$ does not allow a trivial partitioning. We chose to use the factorization of its elements, using the Hopf fibration [52], into two parts, respectively on $S1$ (i.e. an angle in $[0, \pi[$) and $S2$ (the 2-sphere). This particular factorization then allows using a uniform partitioning of $S1$, and any appropriate partitioning of $S2$, which is a less complex problem. We use a triangular mesh [53] that effectively defines bins of similar area and topological shape on the sphere, which is obviously preferable over more trivial methods such as [52]. Finally, let us mention that the size of the bins of the histogram must obviously be chosen in accordance with the actual size of the kernels to be used for the probabilistic evaluation of the density in the second stage of the algorithm.

The paper included in the following pages was presented at the 2012 conference *Digital Image Computing: Techniques and Applications (DICTA)*.

Generalized Exemplar-Based Full Pose Estimation from 2D Images without Correspondences

Damien Teney
University of Liege, Belgium
Damien.Teney@ulg.ac.be

Justus Piater
University of Innsbruck, Austria
Justus.Piater@uibk.ac.at

***Abstract*—This paper addresses the problem of full pose estimation of objects in 2D images, using registered 2D examples as training data. We present a general formulation of the problem, which departs from traditional approaches by not focusing on one specific type of image features. The proposed algorithm avoids relying on specific model-to-scene correspondences, allowing using similar-looking and generally unmatchable features. We effectively demonstrate this capability by applying the method to edge segments. Our algorithm uses successive histogram-based and probabilistic evaluations, which ultimately recover a complete description of the probability distribution of the pose of the object, in the 6 degree-of-freedom 3D pose space, thereby accounting for the inherent ambiguities in the 2D input data. Furthermore, we propose, in a rigorous framework, an efficient procedure for fusing multiple sources of evidence, such as multiple registered 2D views of the same scene. The proposed method is evaluated qualitatively and quantitatively on synthetic and real test images. It shows promising results under challenging conditions, including occlusions and heavy clutter, while being capable of handling objects with little texture and detail.**

I. Introduction and related work

Estimating the pose of a known object in a single 2D image is a fundamental problem in computer vision that has attracted a lot of attention over the years. The task is closely related to the problem of object recognition. However, state-of-the-art object recognition methods usually aims at identifying object *classes*, allowing small variability in appearance among different objects of the same class. We rather focus here on specific *instances* of objects, where such small changes in appearance are actually used as cues for determining the precise pose (3D position and orientation) of the object in a new scene.

The pose estimation task has many direct applications, such as robotic interaction and grasping, augmented reality, or the visual tracking of objects. Methods have been developed that make use of a 3D, explicit geometric model of the object of interest [1], [2], [3]. Those thus require precise a-priori knowledge of the 3D shape of the object, to be provided by external methods such as stereo vision or range sensors. In this paper, we rather present a 2D view-based, or exemplar-based method, which simply uses 2D views of the object as training data, in which the object appears in known poses. Those methods present the advantage of being easily trainable, *directly* using 2D visual data. Further motivation for the exemplar-based approach is brought by the human visual system, which was shown to exhibit properties of a view-based lookup function when recognizing objects, being robust to changes of about 20° around trained viewpoints [4]. Unfortunately, current, state-of-the-art methods following this approach present serious limitations, often relying on specific types of images features, or being suited to only particular types of objects, and are thus able to operate only under limited ranges of conditions. This led us to the reformulation of the problem in more general, probabilistic terms, and to the development of a novel method, that we will introduce after reviewing related work.

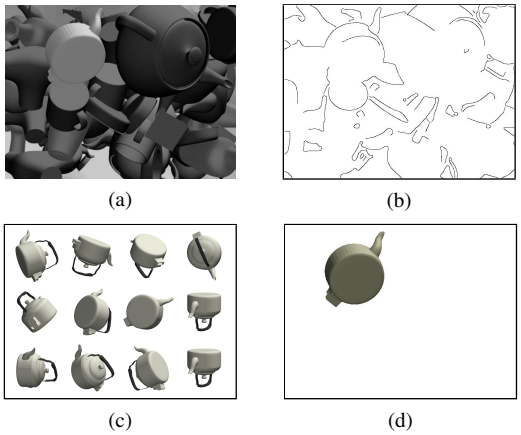


Fig. 1: Pose estimation in a single image, using 2D training examples; (a) test image; (b) edge map used as input; (c) sample training views; (d) rendering of a model of the object in the best pose found by the algorithm, note that the correct pose is recovered despite heavy clutter and missing observations.

Early work in the field of exemplar-based methods used the appearance of the object as a whole. These so-called *appearance-based* methods [5], [6], [7] generally assumed a successful prior detection of the object in the test image and generally offered poor resistance to clutter and occlusions, or did not handle the full 6 degree-of-freedom pose space as needed in practical applications. More recent work, by contrast, focused on the use of individual, precisely located

The research leading to these results has received funding from the European Community's 7th Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience. D. Teney is supported by a fellowship of the Belgian Fund for Scientific Research.

This article is part of Chapter 4: *Histogram on the 6-DoF pose space* (p.41)
It originally appeared in: *2012 Digital Image Computing: Techniques and Applications (DICTA)*

observations (such as *SIFT* features [8]) extracted in the 2D views of the object. These *feature-based* methods [9], [10] then rely on establishing matches, using their appearance, between observations in the test view and in the stored training examples. The limitations of this approach are obviously those of the extraction and matching of image features, which practically works best on texture-rich images, but can perform poorly on scenes with mostly homogeneous surfaces or little detail.

The method proposed in this paper bridges a gap between the two approaches mentioned in the previous paragraph. It makes use of individual features extracted from the images, thereby offering the potential robustness of feature-based methods, e.g. against lighting changes, but does not rely on the matching of specific observations between the test and training views. Practically, this allows using similar-looking types of features. Although the method is generally applicable to different types of observations, we chose to demonstrate this key ability through the use of local edge segments. These correspond to points extracted in the images along the lines of maximum gradient, and they thus carry little appearance information individually. The result of our implementation is a pose estimation method readily trainable with 2D visual data, intrinsically robust to clutter and occlusions, and able to handle previously-problematic objects with little texture and detail.

The identification of the object of interest in a new image ressembles the traditional problem of object recognition and localization. A number of successful methods have been developed that specifically make use of edges as image features. The classical measures of chamfer distance [11] and Hausdorff distance [12] evaluate the fit of a template over a test image; their initial formulations were refined in different ways to provide practical algorithms capable of finding such a template (a training image of the object of interest) in a cluttered scene [13], [14]. One key addition proved to be the use of the orientation of the edges, as we also do in the proposed method. Other state-of-the-art methods include the work of Ferrari *et al*. In [15], they use descriptors of simple edge groupings to train an SVM classifier, capable of recognizing object classes, then using a traditionnal sliding window over the test image. In [16], they focus on the learning of shape models from unsegmented training views, and then use a soft-matching procedure of those shapes to recognize objects in new images. The purpose of those two methods is however to specifically handle intra-class variations of appearance. The work presented in this paper differs from the cited methods in 3 important ways: (i) we present a generally-applicable method not bound to one specific type of image features; it offers the flexiblity to use additional characteristics (e.g. edge curvature) or other features (e.g. interest points); (ii) we do not seek to identify objects or object classes, but rather to determine their pose, using the small changes of appearance as clues to this end; (iii) we go beyond a simple localization in the image (e.g. as 2D bounding box), as we directly consider the full 6-degree-of-freedom pose of the object in the 3D space, of which we recover a probability distribution, and not a single maximum.

The method proposed in this paper is based on a probabilistic representation of both the test and the training data. Such a representation has been used in the slightly different context of pose estimation using 3D models and observations [17], [3], and this work can be seen as their extension to the case of 2D data. In addition to modelling the uncertainty inherent in the input data, the probabilistic approach leads to the definition of the pose of the observed object as a probability distribution in the 3D pose space, of which we want to identify the peaks. This is justified by the uncertainty in the pose estimation problem arising from the 3D-to-2D projection ambiguities. Intuitively, a given 2D view may often be explained by several 3D poses of the object of interest, and we are generally interested in recovering *all* these potentially correct results. Our probabilistic approach, as will be demonstrated, is able to address this objective. Another contribution of this paper is the introduction of successive histogram- and probabilistic-based evaluations that seek to identify *all* significant modes in the distribution of interest. The aforementioned references, which had to deal with less complex distributions, employed approximations such as Monte Carlo methods [18], which generally recovered only a unique solution. This would have been insufficient in the present case, due to the particular ambiguities mentioned above.

Finally, we propose an efficient method for fusing multiple sources of evidence in the same probabilistic framework. This information may be available e.g. through multiple 2D views of the scene, observed under different viewpoints, but the same principle can also serve to jointly handle multiple types of features extracted from a same image. Viksten *et al.* [10] proposed another method for combining such multiple sources of information through a simple clustering step on top of several instances of existing methods. This however lacks the genericity offered by the rigorous approach proposed here. We make full use of the probabilistic nature of the problem, combining the different sources of information in a Markov random field, on which inference is performed using non-parametric belief propagation. The power of the technique is demonstrated through the use of two 2D views of the same scene, thereby increasing the accuracy of the pose estimation process. A comparable approach was used by Toshev *et al.* [19] for tracking of the pose of an object over time in a video. Other methods for handling multiple views with a 2D pose estimation method have been proposed [1], [20], but with the underlying process based on feature matching, as opposed to the more generic approach proposed here.

II. Probabilistic representation of pose and appearance

In this section, we introduce a rigorous formulation of the pose estimation problem, using a probabilistic representation of the input data. As mentioned above, the proposed method is not specific to one particular type of image features, but the general formulation is illustrated with local edge segments.

This article is part of Chapter 4: *Histogram on the 6-DoF pose space* (p.42)
It originally appeared in: *2012 Digital Image Computing: Techniques and Applications (DICTA)*

Those correspond to points extracted from the images along the lines of maximum gradient (see Section V).

A. Representation of test data

Let us first consider the test data, which consists of a single 2D image, from which we extract features x_i. They form the set of *observations* $\mathcal{O} = \{x_i\}_{i=1}^{N}$, where $x_i \in \mathcal{A}$, the space on which is defined the appearance of our observations. In the case of local edge segments, an observation is characterized by its 2D position in the image, and by its orientation (without direction, i.e. an element on the semicircle). Therefore, we have $\mathcal{A} = \mathbb{R}^2 \times S_1^+$. Considering another case where each observation would be a texture patch extracted around an interest point, the appearance space $\mathcal{A}$ would then encompass the position of that point, and a description of the texture itself.

As proposed in [3], such a set of observations can be used to define a continuous probability density ϕ on $\mathcal{A}$. This distribution is defined in a non-parametric fashion, using Kernel Density Estimation (KDE), directly using the elements of $\mathcal{O}$ as supporting particles. The probability density function of ϕ is then given by

$$\phi(x) \;=\; \frac{1}{N} \sum_{x_i \in \mathcal{O}} K_1(x_i, x) \;, \tag{1}$$

where $x \in \mathcal{A}$, and $K_1(\cdot,\cdot)$ a kernel function on $\mathcal{A}$. This formulation allows modelling the uncertainty that may be present in the observations, e.g. due to image noise or to other artifacts occurring during image formation and processing. The kernels used will depend on the appearance space considered [3]. In our application, using edge segments, we found that using kernels allowing only a small deviation on the position and on the orientation was sufficient, as our edge detection algorithm could provide results of good accuracy (see Section V). In practice, the narrow bandwidth of the chosen kernels implies that sampling from $\phi(x)$ amounts to selecting random points x_i from $\mathcal{O}$, with only small variations (see Fig. 2b).

B. Representation of training data

The training data is composed of a number of pre-segmented 2D images, in which the object of interest appears in known poses. Each of those images is processed, in a similar way as the test image, to extract image features. Each observation x_i is then associated with the pose w_i of the image it was extracted from, thereby forming a set of *pose/appearance pairs* $\mathcal{T} = \{(w_i, x_i)\}_{i=1}^{M}$, where $x_i \in \mathcal{A}$, the appearance space of our observations, and $w_i \in \mathrm{SE}(3)$, the space of 3D poses. Similarly to the observations, these points are used to support a KDE, therefore defining a probability distribution on the joint pose/appearance space. This distribution, called ψ, represents the probability of observing an image feature of a given appearance when the object is in a given pose. Formally, ψ is defined by its density function

$$\psi(w, x) \;=\; \frac{1}{N} \sum_{(w_i, x_i) \in \mathcal{T}} K_2\big((w_i, x_i),\, (w, x)\big) \;, \tag{2}$$

where $w \in \mathrm{SE}(3)$, $x \in \mathcal{A}$ and $K_2(\cdot,\cdot)$ is a kernel function on $\mathrm{SE}(3) \times \mathcal{A}$. The use of kernels on the training data can be seen here as a smoothing over the available training points, effectively yielding a continuous distribution and allowing us to interpolate, to some extent, the value of ψ over regions not covered by the training data. Practical details on the use of kernels in $\mathrm{SE}(3)$ are discussed e.g. by Detry and Piater [18].

In addition to the training data, a number of possible transformations in the pose/appearance space are usually known. For example, under orthographic projection[1], the camera intrinsic parameters dictate how a translation (in pose space) parallel to the camera image plane relates to a translation of the observations in the image (in appearance space). In our case, with edge segments, we chose to hard-code three such transformations, namely the translation and rotation in the image plane, and the change of depth along the camera projection rays which give identical projections on the image plane. Formally, we represent these transformations via a single function f, parameterized by a vector of parameters $p \in \mathcal{P}$, such that

$$f\big((w, x),\, p\big) \;=\; (w', x') \tag{3}$$

with (w, x) and (w', x') being pose/appearance pairs, equivalent through the hard-coded transformations under the parameters p. Those transformations allow us to extend our definition of ψ to larger regions of the pose/appearance space than with the training points alone. To that effect, we substitute $\mathcal{T}'$ for $\mathcal{T}$ in Eq. 2, where

$$\mathcal{T}' \;=\; \mathcal{T} \;\cup\; \Big\{ \quad (w', x') : \exists\, (w, x) \in \mathcal{T},\; p \in \mathcal{P} : \\ f\big((w, x),\, p\big) = (w', x') \qquad \Big\}. \tag{4}$$

This *augmented* training set $\mathcal{T}'$ complements $\mathcal{T}$ with all transformations of its elements that can be obtained using f. As we will see in Section III however, our implementation does not require an explicit representation of $\mathcal{T}'$, and, in practice, only a small subset of its elements will have to be identified.

For practical purposes, we remark that the definition of ψ (Eq. 2) presents the problem of making its value dependent on the density of training examples in the corresponding region. For example, including two identical views of the object in the training data, in the same pose, would simply double the density of ψ in the corresponding regions, which is not desirable. This effect is alleviated by using the maximum value of the neighbouring kernels (see Fig. 2c) instead of a summation over their values. This leads to the alternative definition

$$\psi(w, x) \;=\; \frac{1}{C} \max_{(w_i, x_i) \in \mathcal{T}'} K_2\big((w_i, x_i),\, (w, x)\big) \;, \tag{5}$$

where C is a normalization constant.

[1] Our implementation of the method assumes an orthographic or near-orthographic projection, which in practice is easily satisfied with a camera of sufficient focal length relative to the scene depth (see Section V).

C. Probability distribution of 3D pose

The probabilistic representations of the test and the training data, given respectively as ϕ and ψ, are now used together to model the pose of the object in the test image. The pose is modelled as a random variable $W \in \mathrm{SE}(3)$, and its distribution is simply given by

$$p(w) \;=\; \int_{\mathcal{A}} \psi(w,x)\;\phi(x)\;\mathrm{d}x\;. \qquad (6)$$

This expression, in effect, measures the compatibility of a pose w with the whole distribution of features observed in the image. Another interpretation is to see it as the cross-correlation of the distribution ϕ of observations in the test image with the distribution $\psi(w,\cdot)$ of training points at a given pose. Note that this formulation of $p(w)$ is similar to that proposed in [18], [3] for the use of 3D models and observations.

III. Pose inference

This section presents a practical method for solving the pose estimation problem as formulated in Section II. The method is based on two key observations, presented below, which allow an approximate evaluation of $p(w)$.

First, the value of the integral in Eq. 6 can be approximated using Monte Carlo integration [21], [18]. This method, which involves a random exploration of the integration domain, gives

$$p(w) \;\approx\; \frac{1}{n}\sum_{i}^{n}\psi(w,x_i) \quad \text{where} \quad x_i \sim \phi(x)\;. \qquad (7)$$

The evaluation of $p(w)$ (see Fig. 2a–d) thus amounts to successive evaluations of $\psi(w,x_i)$ for different values of x_i, drawn from the distribution of observations in the test image (ϕ).

Importantly, and this is our second key observation, each of these evaluations of $\psi(w,x_i)$ only requires a small number of elements of the *augmented* training set $\mathcal{T}'$. For a fixed x_i, using the hard-coded transformations (in-plane rotation and translation), any original training pair $(w,x) \in \mathcal{T}$ can be transformed into a pair $(w',x_i) \in \mathcal{T}'$ of appearance x_i. Those pairs will have the strongest influence on the value of $\psi(w,x_i)$ (Eq. 5), and its evaluation can therefore be limited in practice to the use of those pairs, which formally correspond to the following subset of $\mathcal{T}'$:

$$\{\quad (w',x_i) : \exists\,(w,x)\in\mathcal{T},\; p\in\mathcal{P} : \\ f\big((w,x),\;p\big) = (w',x_i) \qquad \} \;\subset\; \mathcal{T}' \qquad (8)$$

The practical consequence of this property is that an explicit and complete representation of $\mathcal{T}'$ is not required, and that only a fraction of its elements have to be identified.

A. Exhaustive search algorithm

The two properties we just presented make the evaluation of $p(w)$ possible for any pose w. Various methods can then in principle be used to identify the main modes of this distribution, such as a Monte Carlo-type search as proposed in [18], [3]. However, the purpose of such methods is generally to identify the global maximum of the density. As argued above, the particular ambiguities in the 2D input data are likely to induce a very complex distribution, potentially presenting multiple weak modes that we wish to identify. We therefore devised an algorithm to exhaustively explore the relevant parts of the 3D pose space. This task is particularly challenging [22] due to the high dimensionality of SE(3). We propose a two-stage process that first relies on a histogram-based approximation, in order to pre-select regions of interest in SE(3). This serves to discard those bins of the histogram that correspond to areas of low density, dramatically reducing the amount of data used at the second stage. It is then possible to perform a full-scale kernel-based evaluation of the density (Eq. 5,7), limited to the pre-selected regions of the pose space. The algorithm returns a set of poses $\mathcal{R}$ where the density exceeds a certain threshold.

The computational complexity of this algorithm is proportional to the number of training points (M, Section II-B), multiplied by the number of samples used from the observations (n, in Eq. 7), itself chosen as a fraction of the total number of observations (N, Section II-A). Note also that it is not mandatory to process all possible combinations of observation samples and training points, but a stochastic approach can rather make use of the probabilistic representation of the training data ϕ, and use a limited number of random samples thereof. This scheme was previously used in the related problem of 3D models and observations [17], [3].

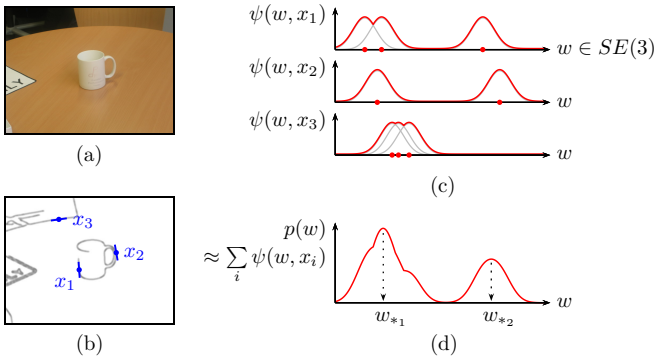


Fig. 2: Proposed method for pose estimation, using edge segments as image features. (a) Test image; (b) distribution of edges in test image, denoted $\phi(x)$ in the text, and three samples x_{1-3} (*blue oriented points*) of that distribution; (c) distribution (*red curve*) of poses compatible with each observation x_i (Eq. 5), made up of individual kernels (*gray curves*) supported by a small subset of poses $w' \in \mathcal{T}'$ (Eq. 8, *red dots*); (d) distribution of poses compatible with all observations x_i (Eq. 7) and local maxima w_{*_i}, as recovered by our method.

B. Post processing of pose estimates

As a post-processing step, one may want to identify the actual peaks of each mode. This could be accomplished by a traditional gradient-ascent method, such as mean shift

[23]. In our case, this procedure would be costly due to the complexity of the pose space. Fortunately, in practice, the proposed algorithm usually returns poses in the close neighbourhood of the actual peaks. A simple *non-maximum suppression* step therefore proves sufficient. In this method, an element is discarded if it lies in the close neighbourhood of an element of greater density, the neighbourhood being defined by a fixed radius in the pose space. This procedure, efficiently implemented by processing the poses of $\mathcal{R}$ in order of decreasing density, therefore selects the poses that are the closest to the peaks of the distribution (Fig. 2d).

IV. Extension to multiple sources of evidence

The method presented above uses a single source of information as input data, i.e. a single 2D image, to evaluate the most probable poses of the object. However, it is sometimes desirable to use several sources of information to disambiguate the result, or make it more precise. Such extra information could be available, e.g. as multiple images of the same scene, observed under different viewpoints, or as several types of image features, extracted from one same image. This section proposes a rigorous method for fusing the results produced by each different cue, thereby determining globally consistent poses. The method is presented in the concrete context of multiple views, but it directly extends to other scenarios, e.g. with multiple types of image features.

We represent by the random variable $W \in \mathrm{SE}(3)$ the pose of the object, and by $X_i \in \mathcal{A}$ the distribution of observations in the ith view. The dependency between these random variables can be represented by a pairwise Markov random field [24], [17], organized in a tree structure, W being the root node (see Fig. 3). The *compatibility potential functions* parameterizing the relationship between W and a X_i are called ψ_i. These are identical to the ψ introduced in Section II, apart from now taking into account the actual viewpoint of the corresponding view. Each node X_i is moreover connected to its corresponding observed variable, Y_i , their relationship being parameterized by ϕ_i, defined similarly to the ϕ of Section II. To determine the marginal density of the top node W, inference on such a graphical model can be performed using Non-parametric Belief Propagation (NBP), as proposed in [24]. The application of the NBP algorithm on a model as simple as that considered here allows many simplifications. In particular, the distribution of W is simply given by

$$p(w) \;=\; \prod_{i=1}^{q} m_i(w) \;, \tag{9}$$

with a *message* $m_i(w)$, conceptually sent from a node X_i to the root node W (see Fig. 3), and expressing its *belief* about the state of W, being defined as

$$m_i(w) \;=\; \int_{\mathcal{A}} \psi_i(w,x)\,\phi_i(x)\,\mathrm{d}x \;. \tag{10}$$

Note that this definition of $m_i(w)$ is identical to Eq. 6, but is now indexed on the source of the observations. Practically, each $m_i(w)$ can be independently evaluated, using the method

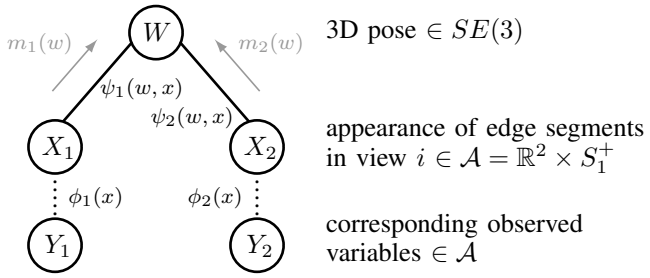


Fig. 3: Markov tree representing the integration of multiple source of evidence, in this case 2 views from which edge segments are extracted. The messages $m_i(w)$ represent the belief about the state of W sent from each node X_i; they are fused to determine the values of W globally consistent with the two views.

of Section II-C. This method returns a set of poses in the most dense regions of SE(3), which can directly be used to represent the distribution $m_i(w)$ in a non-parametric fashion, using KDE, weighting each of them with its evaluated probability density. Fusing the results from all sources of evidence, via Eq. 9, then amounts to computing the product, or intersection, of all of these non-parametric representations of densities on SE(3). In practice, the representation of each $m_i(w)$ is usually quite compact, and the evaluation of $p(w)$ for a given w can thus be performed at a reasonable computational cost. We therefore identify the maxima of $p(w)$ with a Markov Chain Monte Carlo (MCMC) type search, using a simple random walk scheme [18]. This local optimization process is performed from several different starting points, selected from the supporting particles of the $m_i(w)$. Using this process, the output of the algorithm is finally the set of poses corresponding to the local maxima of $p(w)$ as defined by Eq. 9, i.e. the poses globally the most consistent with all available sources of evidence.

V. Evaluation

The evaluation of a pose estimation method is not trivial, due to the difficulty of obtaining the ground truth 3D poses themselves, especially in realistic scenes. We considered using various existing datasets, as reviewed below, but finally decided to produce new datasets, with synthetic images and thus known ground truth, which allowed performing a rigorous quantitative evaluation. Practically, the image features were extracted using the well-known Canny edge detector, followed by a smoothing and subsampling step to reduce the noise in the observations (Fig. 1b). All images used were 640×480 pixel grayscale images, and all the parameters of the algorithm were set to identical values for all the tests (with both synthetic and real images).

Among candidates public datasets, we considered the *ETHZ Shape dataset* [15], which features shape-based object classes in various cluttered scenes. It is however specifically targeted at class recognition algorithms, designed to handle variations in shape, as opposed to our method, which actually uses

those slight changes in appearance as clues for estimating the 3D pose of the object. The dataset does not include any suitable training data or any ground truth for 3D pose. The *NORB dataset* [25] is made up of images of toy objects in different poses, and of artificial compositions of such images proposed as cluttered scenes. In addition to being evaluated only with class recognition methods (as far as we are aware), the very-low-resolution images prevent any reliable use of edge features, as our method requires. The *RGB-D dataset* [26] is made up of household objects on a turntable, viewed at 3 different elevations, thus in a fairly limited range of poses. We also argue that the basic evaluation methodology proposed for those sequences, which is basically to use every other image for training and test alternatively, in the absence of clutter and object translations, is overly simplistic and of limited diagnostic value. The capture setup (e.g. constant-speed turntable) is also acknowledgedly imprecise and ruled out this dataset as an interesting candidate for a rigorous evaluation.

A. Quantitative evaluation on synthetic images

The synthetic datasets were produced with manually designed 3D models and rendered with ray tracing software. The training examples (Fig. 1c) correspond to different views of the object of interest on a uniform background; the poses of the object in the training set are chosen uniformly in the orientation space. The amount of clutter in a test image is measured as the ratio of the number of observations *not* belonging to the object of interest over the total number of observations in the image. For example, a clutter ratio of 0% corresponds to absence of clutter, whereas a clutter ratio of 80% means that about $4/5$ of the observations are actually noise. We measure the success rate as the ratio of experiments that returned a *correct* pose in the first k results (the algorithm returns a list of poses sorted by decreasing probability density). This aspect is important, as the ambiguities the 2D input data often prevent one from distinguishing between different 3D poses that have very similar appearances on the image plane. The threshold for considering a pose as *correct* was set in accordance to the typical dimensions of a scene: considering our objects are of a size of 100–200 mm and distant from the camera of 1000–2000 mm, this threshold was set to a translation error of 20 mm parallel to the image plane (XY), 100 mm in depth (Z), and a maximum rotation error of $20°$. The greater tolerance on the Z translation is justified by the fact that the use of a single 2D image makes the determination of depth very difficult. Note however that this error threshold remains a small fraction of the actual depth of the scenes. Using these conventions, the success rates of the algorithm for various conditions are reported in Fig. 5a. Please also note that relaxing the threshold discussed above does not necessarily lead to better quantitative results, as we also report, in Fig. 5c, the mean error of the first *correct* result returned by each run of the algorithm. The reported average numbers were computed over 30 runs of the algorithm for each of the 6 objects considered (Fig. 5b), each scene being generated at random, with clutter made up of different objects disposed randomly in the background. The measure of the error in orientation for the cylindrically symmetric objects (e.g. the bottle) naturally takes only their relevant degrees of freedom into account.

Systematic test cases including occlusions are hard to design, as the amount of occlusion is difficult to quantify: masking one half or the other of an object can have dramatically different effects due to different levels of detail. We are however confident in the ability of the system to cope with significant occlusion, since this is actually simulated by a common large fraction of missing observations (Fig. 4), due to background clutter preventing a good extraction of edges.

The algorithm presents very good success rates under common amounts of clutter (Fig. 5a). This success rate even remains acceptable as the amount clutter is raised to very challenging values (Fig. 4). Increasing the number of training views for each object was not found to have a significant impact on the success rate, but increased the accuracy of the results (Fig. 5c). Similarly, the amount of clutter did not have a significant influence on the precision of the results (Fig. 5c), but only makes harder the identification of the modes of the distribution. In general, the erroneous results can be attributed to two sources (see Fig. 4, last row). First, the edge segments we restrict ourselves to cannot always be extracted consistently. For example, in an image of the kettle, if the edges of the handle are extracted on one of its sides but not on the other, this side may be "matched" with any of the two sides of a training view, potentially leading to a large error on the orientation of the recovered pose – despite both being globally good matches with the 2D input view. Second, using the 2D projections of any 3D object introduces inevitable ambiguities. For example, it may be very difficult to differentiate between a cylindrical object pointing away and towards the camera (Fig. 4, bottom left); this effect is particularly true for our objects consisting of mostly homogeneous surfaces.

We used a similar protocol to evaluate the use of multiple views of a same scene, as proposed in Section IV. In those experiments, we used, instead of a single 2D image, 2 images of the scene from viewpoints spaced by $45°$. Such a wide baseline is generally too large to be handled by traditional stereo methods, and thus demonstrates one of the interests of our approach. The success rate was generally not noticeably affected by the use of two views over one, but the error was almost always substantially decreased, as reported in Fig. 5c. Using a second view helps the algorithm disambiguating between the different possible orientations of the object, and also provides much better clues for determining the actual depth of the scene (Z translation).

B. Real test images

The method was evaluated on real test images. For practical reasons, we relied here again on computer-generated images as training data. We used 128 training views of each object, that were produced as explained above (Fig. 1c), through ray tracing with manually-designed 3D models. In a realistic appli-

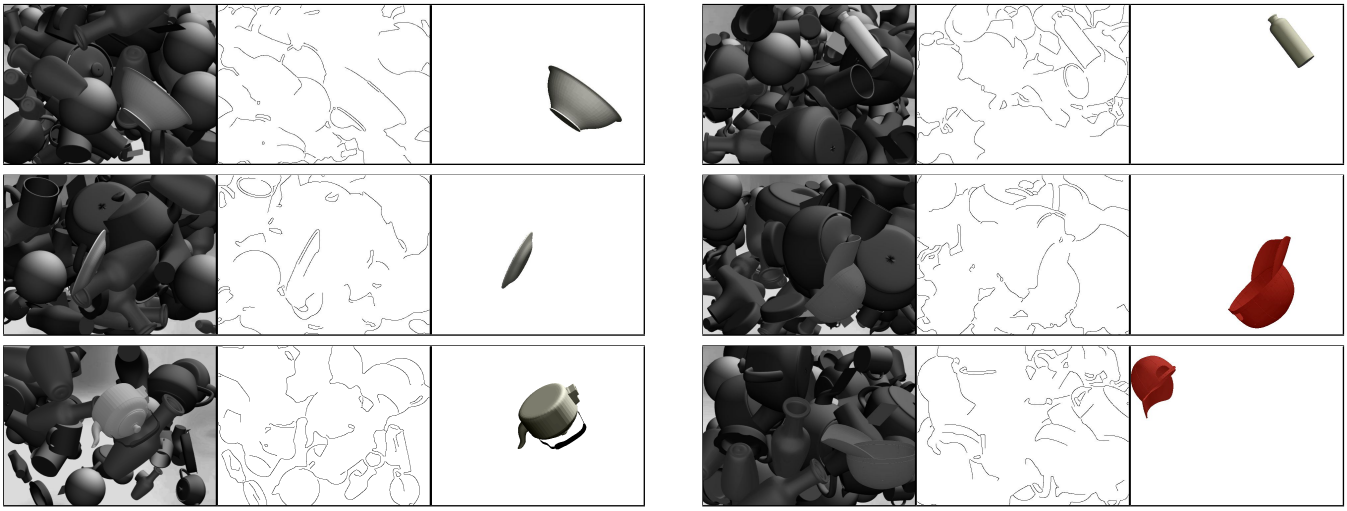
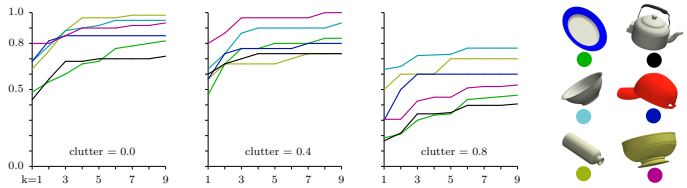


Fig. 4: Sample results of our quantitative evaluation (for each: test image, edge map used as input, rendering of object model in the first pose proposed by the algorithm); these tests used a single test view, 128 training views per object, and clutter=80%. The last row shows typical incorrect results: although a close match is found with the given edges, the 3D pose is incorrect.



(a)

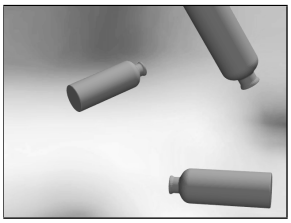
(b)

N. of train. examples	**32**	**128**	**512**	**128, stereo obs.**
Translation (mm)	44.1	30.5	31.1	14.2
XY only	4.6	3.1	2.1	N/A
Z (depth) only	43.2	30.1	30.9	N/A
Orientation (°)	11.6	8.3	7.7	8.1
Rank (k)	2.9	1.8	1.3	1.2

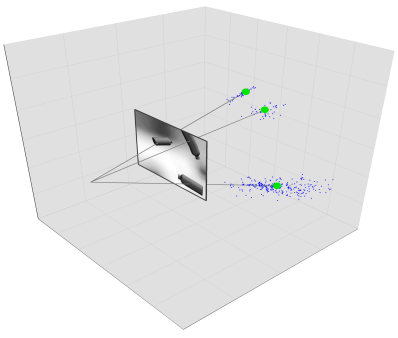
(c)

Fig. 5: Quantitative results on synthetic images (see text for details); (a) for each object, success rate of having a correct result among the first k ones (128 training examples), in scenes of no/medium/heavy clutter; (b) test objects used; (c) average error of first correct result.

Fig. 6: Sample results on real images (similar conditions as Fig. 4); for visualization, we render, in yellow, the outline of artificial models set in the first pose found. The last two images show common failures, typically due to uncertainty in the limited input data used (edges): the mitten identified in background clutter, and the rim of the plate matched onto its shadow.



(a)



(b)

Fig. 7: Recovery of probability distribution of 3D pose; (a) input image; (b) plot of 3D position as a non-parametric description (blue points), and local maxima (green points). Each occurrence of the object in the image correctly generates one mode in the distribution.

cation, such images are to be acquired, e.g. by a robotic agent taking pictures of the real object under various viewpoints [27]. This alternative option was chosen purely for practical reasons, but added an additional challenge as the models used for generating the training images inevitably did not match the real objects perfectly. The test images were taken with a handheld camera at about 1000–2000 mm from the scenes.

We performed many experiments on typical household scenes with common objects. We purposely chose objects presenting large homogeneous surfaces with little texture and details, on which classical feature-based pose estimation would likely fail. We present, in Fig. 6, typical results of both successful and failed experiments. As the ground truth pose is not available, measuring the errors is not possible. Instead, we visualize the results by rendering, onto the input images, synthetic models of the objects in the poses found by the algorithm. One can observe good matches with the input images, demonstrating the good use made of the 2D information available. As discussed before, the use of 2D observations, especially edge segments alone, often makes it hard to distinguish between different poses that may appear similar in one image. The first pose returned by the algorithm may thus correspond to an erroneous result, but the correct result will often be found in the other poses proposed by the algorithm (identified with slightly lower probability). The actual disambiguation is thus to be left to the end application, which may best make use of this uncertainty in the results.

C. Retrieval of full pose distribution

One key capability that we propose is to recover a *distribution* of 3D poses, rather than a single optimum. We illustrate this in Fig. 7: the pose of a bottle is evaluated in an image containing several occurrences of the object. The distribution is recovered in a non-parametric fashion as a collection of particles, of which we plot the 3D position. One mode is correctly identified for each occurrence of the object, the main uncertainty remaining unsurprisingly in the depth dimension, extending along the camera projection axis.

VI. Conclusions and future work

We presented a novel method for exemplar-based pose estimation in single images. Relying on a general, probabilistic formulation of the problem, the method avoids establishing specific correspondences between training and test views, thus allowing similar-looking types of images features. The pose of the object is treated as a probability density over the 3D pose space, from which we identify the different modes, thereby accounting for the ambiguities of 2D input data. We also proposed an elegant way of fusing evidence from multiple sources, such as several views of the same scene, or different types of image features. A first validation of the overall approach showed promising results. Further developments will mainly focus on the use of other types of image features within this framework, extending its applicability further to more types of scenes, objects and imaging conditions.

References

[1] A. Collet and S. S. Srinivasa, "Efficient multi-view object recognition and full pose estimation," in *IEEE Int. Conf. Rob. and Autom.*, 2010, pp. 2050–2055.

[2] I. Gordon and D. G. Lowe, "What and where: 3D object recognition with accurate pose," in *Toward Category-Level Object Recognition*, 2006, pp. 67–82.

[3] D. Teney and J. Piater, "Probabilistic Object Models for Pose Estimation in 2D Images," in *DAGM*, ser. LNCS, vol. 6835. Springer, 2011, pp. 336–345.

[4] S. Edelmann and H. Bülthoff, "Modeling human visual object recognition," in *Int. Joint Conf. Neural Networks*, 1992, pp. 37–42.

[5] S. Ekvall, F. Hoffmann, and D. Kragic, "Object recognition and pose estimation for robotic manipulation using color cooccurrence histograms," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2003.

[6] P. Mittrapiyanuruk, G. N. DeSouza, and A. C. Kak, "Calculating the 3D-pose of rigid-objects using active appearance models," in *IEEE Int. Conf. Rob. and Autom.*, 2004, pp. 5147–5152.

[7] A. R. Pope and D. G. Lowe, "Probabilistic models of appearance for 3D object recognition," 2000.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints." *Int. J. Comp. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[9] R. Soderberg, K. Nordberg, and G. Granlund, "An invariant and compact representation for unrestricted pose estimation," in *Patt. Rec. and Im. Anal.*, J. Marques, N. Prez de la Blanca, and P. Pina, Eds. Springer, 2005, vol. 3522, pp. 489–500.

[10] F. Viksten, P.-E. Forssén, B. Johansson, and A. Moe, "Comparison of local image descriptors for full 6 degree-of-freedom pose estimation," in *IEEE Int. Conf. Rob. and Autom.*, 2009, pp. 2779–2786.

[11] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 849–865, 1988.

[12] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, 1993.

[13] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *Conf. Comp. Vis. and Patt. Rec.*, 2010, pp. 1696–1703.

[14] C. F. Olson and D. P. Huttenlocher, "Automatic target recognition by matching oriented edge pixels," *IEEE Trans. Im. Proc.*, pp. 103–113, 1997.

[15] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Object detection by contour segment networks," in *European Conf. Comp. Vis.*, 2006.

[16] V. Ferrari, F. Jurie, and C. Schmid, "From images to shape models for object detection," *Int. J. Comp. Vis.*, vol. 87, no. 3, pp. 284–303, 2010.

[17] R. Detry, N. Pugeault, and J. Piater, "A probabilistic framework for 3D visual object representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1790–1803, 2009.

[18] R. Detry and J. Piater, "Continuous surface-point distributions for 3D object pose estimation and recognition," in *Asian Conf. Comp. Vis.*, 2010.

[19] A. Toshev, A. Makadia, and K. Daniilidis, "Shape-based object recognition in videos using 3d synthetic object models," in *Conf. Comp. Vis. and Patt. Rec.*, 2009, pp. 288–295.

[20] F. Viksten, R. Soderberg, K. Nordberg, and C. Perwass, "Increasing pose estimation performance using multi-cue integration," in *IEEE Int. Conf. Rob. and Autom.*, 2006, pp. 3760–3767.

[21] R. Caflisch, "Monte carlo and quasi-monte carlo methods," *Acta Numerica*, vol. 7, pp. 1–49, 1998.

[22] R. C. Nelson and A. Selinger, "Large-scale tests of a keyed, appearance-based 3D object recognition system," *Vis. Res.*, vol. 38, pp. 38–15, 1998.

[23] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790 –799, 1995.

[24] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *Comp. Vis. and Patt. Rec.*, 2003.

[25] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Conf. Comp. Vis. and Patt. Rec.*, 2004, pp. 97–104.

[26] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *IEEE Int. Conf. Rob. and Autom.*, 2011.

[27] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann, "Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot," in *IEEE Int. Conf. Rob. and Autom.*, 2010, pp. 2012–2019.

Chapter 5

Extension from specific objects to object categories

This chapter presents a series of improvements to the method introduced in the previous chapter. The central one is the extension from specific objects to object categories. In this case, training images are available for a number of different objects, and serve to learn a common model, representative of the appearance of a category defined by all those objects. The category is thus defined implictly by the examples provided as training data. We reuse our nonparametric representation of probability distributions of image features, with, this time, features from *multiple* training images together, instead of one image at a time. One obvious limitation worth mentioning here is the fact that we do not take into account the co-occurrence of features in the training examples. This means that the resulting model can represent all combinations of variations present in these examples. A model learned from cars and giraffes would not only represent these two types of objects, but also anything looking part-car and part-giraffe. This can be seen as a strength, as few examples can suffice to represent wide variations of overall appearance. However, this also means that the overall procedure will practically be most effective with training examples sharing strong visual characteristics, not e.g. with categories defined semantically and including instances looking vastly different.

The algorithm to solve for the optimal pose (including the localization in the image) is similar to the one of Chapter 4, but has been simplified to become less expensive computationally. Instead of casting votes in the full (6-degrees of freedom) pose space, we now iterate over discrete values for the dimensions corresponding to the scale, image (in-plane) rotation, and viewpoint. Votes are thus cast only on the 2 remaining dimensions, which correspond to the localization in the image. Interestingly, the resulting algorithm is then akin to the well-known Generalized Hough Transform [27], which is a classical method to perform detection of rigid templates in images. Our method now thus explicitly considers each training viewpoint independently at a time. The capability for continuous pose estimation is provided by a second step, operating on top of these detections: considering the measure of similarity, or "matching score" between the test image and each of the training views, we obtain an "average"

of these detections by fitting a distribution over the similarity scores, and by retaining the mean of its main mode. This simple procedure leads to remarkable results in terms of accuracy of the estimated pose. We obtained results superior to the state-of-the-art on the "rotating cars" dataset [54].

This chapter also presents a way of weighting the training data to further improve the accuracy in pose estimation. As noted by several authors in earlier work [10, 23, 55, 56], such a weighting can indeed significantly improve results of voting-based image localization. The basic idea is to assign heavier weights to the training features that are the most informative to the task at hand. In our specific implementation, since we focus on pose estimation, we increase the relative weights of the features that unambiguously characterize the appearance of the object at specific viewpoints.

The paper included in the following pages was presented at the 2013 *Computer and Robot Vision* conference, where it received the "Best Vision Paper" award.

Continuous Pose Estimation in 2D Images at Instance and Category Levels

Damien Teney
University of Liège, Belgium
Damien.Teney@ULg.ac.be

Justus Piater
University of Innsbruck, Austria
Justus.Piater@UIBK.ac.at

***Abstract*—We present a general method for tackling the related problems of pose estimation of known object instances and object categories. By representing the training images as a probability distribution over the joint appearance/pose space, the method is naturally suitable for modeling the appearance of a single instance of an object, or of diverse instances of the same category. The training data is weighted and forms a generative model, the weights being based on the informative power of each image feature for specific poses. Pose inference is performed through probabilistic voting in pose space, which is intrinsically robust to clutter and occlusions, and which we render tractable by treating separately the least interdependent dimensions. The scalability of category-level models is ensured during training by clustering the available image features in the joint appearance/pose space. Finally, we show how to first efficiently use a category-model, then possibly recognize a particular trained instance to refine the pose estimate using the corresponding instance-specific model. Our implementation uses edge points as image features, and was tested on several existing datasets. We obtain results on par with or superior to state-of-the-art methods, on both instance- and category-level problems, including for generalization to unseen instances.**

I. Introduction and related work

The problem we focus on is the localization and the estimation of the precise 3D pose of objects in a new scene, given a single image of that scene, and multiple images of the objects as training examples. This is a central problem in computer vision, and there exists a wealth of literature on the topic, especially when dealing with specific object *instances*, e.g. a particular car or a particular coffee mug. The classical methods rely on the use discriminative image features and descriptors (such as SIFT or Geometric Blur), matched between the test view and the training examples. Such features are sometimes stored together with a rigid explicit 3D model of the object [1], [2], which brings viewpoint-invariance to the model. Other techniques have been proposed to encode viewpoint-invariant models, especially in the context of object recognition, e.g. by linking the observed features across different viewpoints [3], [4], [5], or modeling the object as a collection of planar parts [4]. Those methods however were used mainly with the goal or *localizing* and *recognizing* those objects in the images, but without recovering their 3D pose explicitly. One exception is the work of Savarese *et al.* [4], but the recovered pose is only a rough identification, such as "frontal view" or "side view". This limitation is present in many other methods [6], [7], [4], [8] which use discretized pose values, treated as separate classes, with different classifiers tuned to each of them. There exist however methods, often presented in the robotics community (with applications such as object grasping in mind), which can provide accurate pose estimates [9], [10], but they are mostly limited to specific object instances.

One particular aspect we are interested in is to provide the capability for pose estimation at the *category* level. There is an increased interest for this more challenging task; the goal is for example to train the system with a set of different mugs, then to recognize the pose of a new, unseen mug. The categories in such a scenario are defined implicitly by the training instances used as examples.

Previous work on object recognition does acknowledge the close link between handling the variability of object appearance as a function of pose and due to the diversity of objects within a category. Gu and Ren [11] showed how to solve for instance and *discrete* (coarse) pose recognition at the same time. Lai *et al.* [12] did so as well, using a tree of classifiers tuned for the different tasks. However, they use presegmented views of the objects, without any clutter or occlusions, and provide modest results on the accuracy of the retrieved pose. The methods mentioned in the previous paragraphs, while modeling the change of appearance due to different viewpoints, generally cannot directly handle the variability within *categories* of objects [3], [5]. One way this capability has been provided is by encoding — in addition to a rough 3D model — the possible variations in appearance [13], [14]; one limitation however is that no shape variability is possible. Our model, on the contrary, is purely appearance-based, and naturally accommodates variability in shape as well as in appearance. The traditional models of rigid geometrical constraints and highly discriminative features [2] are not adequate for encoding within-category variations. One exception to most methods here is again the model of Savarese *et al.* [4], which is specifically designed to provide viewpoint-invariance while handling within-category differences — but still provides only coarse pose estimates.

Recently, some methods have been introduced that can handle category variability and perform localization together with *precise* pose estimation. Glasner *et al.* [15] uses structure-from-motion to reconstruct accurate 3D models from the training images. They then account for within-category variability simply by merging multiple exemplars in their non-parametric model, in a fashion very similar to us. They perform pose infer-

This article is part of Chapter 5: *Extension from specific objects to object categories* (p.51)
It originally appeared in: *2013 Computer and Robot Vision (CRV)*

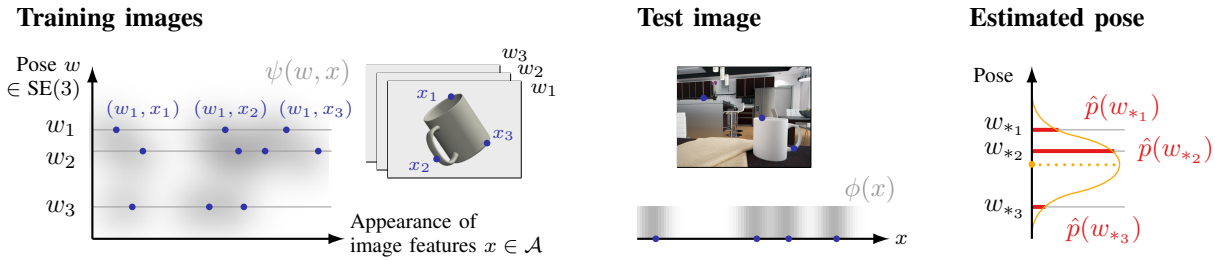


Fig. 1: Proposed method for representing training/test data and for pose estimation. Images features (blue points) are extracted from training and images; their appearance descriptor (in the case of our edge points, a position and orientation in the image) is defined on the generic space $\mathcal{A}$. Training/test observations define, using KDE, continuous probability distributions, respectively ψ and ϕ (gray shaded areas). Our pose inference algorithm (Fig. 2) returns approximations (red bars) of the pose likelihood function $p(w)$ at some discrete poses w_{*_i}. Finally, we locally fit, on those approximations, a simple distribution in the pose space (orange curve), and keep its mean as our final, precise pose estimate (orange dot).

ence through probabilistic voting in the 6D pose space, again in a similar way as we do, thereby solving for localization and viewpoint identification together. However, the reconstruction of such dense 3D models relies on the initial availability of a large number of views. By contrast, the appearance-based model used in this paper can use an arbitrary number of views and can be incrementally updated as more views become available. In a very different approach, Torki and Elgammal [16] learn a regression from local image features to the pose of the object. They recover a precise pose, but cannot handle significant clutter or occlusions, and the accurate pose estimation depends on the (supervised) enforcement of a one-dimensional manifold constraint (corresponding to the 1D rotation of the object in the training examples). It is not clear how that approach would extend to the estimation of the full 3D pose of an object. On the contrary, our method is framed from the start in the context of the full 3D pose.

Our method can accommodate different types of image features, but we chose to use very basic points along edges (combined with their tangent orientation) as opposed to more elaborate features such as SIFT descriptors. Recognition by matching such descriptors, while easier with specific instances, does not easily extend well to object categories. We differ from most edge-based shape recognition methods (e.g. [17], among many others) by avoiding intermediate representations such as contour fragments, and leveraging the simplicity of low-level features — in our implementation, simple points along edges. These simple features provide invariance to viewpoint and to within-category changes of appearance. Using such non-discriminative features for recognition however raises an additional challenge, since no matching is possible. This motivated the use of the framework proposed by Teney and Piater [18] for pose estimation in 2D images, which does not rely on correspondences between the test and the training data. Like [4], this model is generative and does not include any discriminative aspects, but has however been shown to be useful for localization and recognition in the presence of heavy clutter and occlusions [18]. Compared to that work, (1) we use a more efficient method for pose inference that does not need to consider the whole 6D pose space at once, (2) we introduce a weighting scheme of the training features which, as we will show, enhances significantly the performance of the system, and (3) we extend the methodology from instance-specific to category-level models.

The capabilities of the approach proposed in this paper differ from existing work by (1) handling, within the same framework, *instance*-specific models and *category*-specific models of objects, in the latter case allowing variations in shape and appearance, (2) performing *continuous* (precise) 3D pose estimation using those models, as opposed to viewpoint classification and coarse pose estimates, and (3) using such models to solve pose estimation *and* image localization together, as opposed to competing methods that do not handle clutter or occlusions. In addition, we present how to use category- and instance-level models successively, for optimal accuracy and efficiency: the category-model is used first to recover an initial pose estimate, which then allows one to possibly recognize a particular trained instance, so that the corresponding instance-specific model can be used to refine the pose estimate. Finally, in Section IV, the performance of our approach is compared to the most closely related methods [7], [4], [16]; we obtained promising results, on par with or superior to published data.

II. Pose estimation of specific object instances

A. Probabilistic representation of input data

The method we use is based on a probabilistic representation of both the training and the test data. This approach can be seen as a smoothing over the available data, providing continuous distributions of features and interpolating, to some extent, between the available data points (see Fig. 1, left and middle). Practically, the training examples are a set of K images of the object to learn, each annotated with the 3D pose of the object, $w_k \in \mathrm{SE}(3)$ with $k = 1, \ldots, K$. We extract, from each training image, features x_i, which are edge points (see Section IV) with their tangent orientation, and which are thus defined on $\mathbb{R}^2 \times S_1^+$ (accounting for the position in the image, plus an orientation without direction). In the general case, we will call this space the *appearance* space, $\mathcal{A}$. We then

pair all features x_i of a view k with the pose w_k, so that we obtain a set of *pose/appearance pairs* $(x_i, w_k)_i$. Considering the whole training set, the pairs from all example images are concatenated to form our full training set $\mathcal{T} = \{(w_i, x_i)\}_{i=1}^{M}$, with $x_i \in \mathcal{A}$, and $w_i \in \mathrm{SE}(3)$.

The elements of our training set are then simply used to define a continuous probability distribution ψ on the pose/appearance space, in a non-parametric manner, with kernel density estimation:

$$\psi(w, x) \;=\; \frac{1}{M} \sum_{(w_i, x_i) \in \mathcal{T}} \mathsf{K}_1(w, w_i)\, \mathsf{K}_2(x, x_i)\;, \qquad (1)$$

where $w \in \mathrm{SE}(3)$ and $x \in \mathcal{A}$. The kernel functions $\mathsf{K}_1(\cdot,\cdot)$ and $\mathsf{K}_2(\cdot,\cdot)$ handle respectively the pose and the appearance spaces. Details on suitable kernels can be found, e.g. in [18], [19]; the first is an isotropic kernel allowing small deviations in both position and orientation, and the second, similarly, allows small variations in the location in the image and tangent orientation of the image feature.

The test data, which is a single 2D image of a new scene, is handled in a similar fashion as the training data. We extract the same type of image features, which we store as a set of *observations* $\mathcal{O} = \{x_i\}_{i=1}^{N}$, where $x_i \in \mathcal{A}$. This set is then used to define the continuous probability density ϕ on $\mathcal{A}$:

$$\phi(x) \;=\; \frac{1}{N} \sum_{x_i \in \mathcal{O}} \mathsf{K}_2(x, x_i)\;. \qquad (2)$$

As noted in [18], the transformations in the pose/appearance space corresponding to in-plane rotations/translations/scale changes are known from the camera calibration; those trivial transformations (e.g. a change in depth corresponds to a change of scale) are thus hard-coded. This allows us, when using ψ as a generative model, to extend its definition to parts of the pose space not explicitly covered by the training data.

B. Pose inference

The pose of the object of interest in the test scene is modeled as random variable $W \in \mathrm{SE}(3)$, the distribution of which is given by the likelihood function

$$p(w) \;=\; \int_{\mathcal{A}} \psi(w, x)\, \phi(x)\, \mathrm{d}x\;, \qquad (3)$$

This expression simply measures the compatibility of the training data at a pose w, with the distribution of features observed in the test image. The objective is to identify the main modes and peaks of the distribution of W, which was accomplished in [18] by a probabilistic voting scheme on the 6D pose space. This procedure is extremely costly in memory and processing [15], [18] due to the high dimensionality of the pose space. We now propose an approximation of that method that handles different dimensions of the pose space in different ways. Formally, a pose $w \in \mathrm{SE}(3)$ can be decomposed as a concatenation of 3 simpler entities, such that $w = w^3 \circ w^2 \circ w^1$. The first, w^1, corresponds to the "viewpoint", i.e. which side of the object is facing the camera; w^2 is a combination of an in-plane rotation and scale change, and w^3 corresponds to a pure translation parallel to the image plane. The main supporting observation for our proposed method is that a significant peak in the distribution of W will most likely appear as a peak in the distribution corresponding to the dimensions of w^3 alone. Indeed, an object of the test scene in any specific pose w will appear at a *precisely defined* image location (dimensions of w^3). This leads to the algorithm presented in Fig. 2, which iterates over discretized values for the dimensions of w^1 and w^2, and uses probabilistic voting only on the dimensions of w^3 (the 2D localization in the image). The peaks in those last two dimensions are thus identified by the algorithm for discrete viewpoints, scale and in-plane rotation values. This formulation is reminiscent of the classical Hough voting scheme used extensively for object localization [20]. The main advantage over [15], [18] is to avoid considering the entire pose space at once.

Input: training pairs $\mathcal{T} = \{(w_i, x_i)\}_i$ defining ψ
test observations $\mathcal{O} = \{x_i\}_i$ defining ϕ
Output: set $\mathcal{R}$ of approximations of the pose likelihood function
$\mathcal{R} = \{(w_{*_i}, \hat{p}(w_{*_i}))\}_i$

Procedure:
$\mathcal{R} \leftarrow \emptyset$
For each discrete w^1 in $\mathcal{T}$ (viewpoint)
  For each discrete step of w^2 (in-plane rotation and scale)
    Considering pose $w' = w^2 \circ w^1$,
    find best w^3 (image translation) between $\psi(w', x)$ and $\phi(x)$:
      Get samples: $(w_i^\psi, x_i^\psi) \sim \psi(w', x)$
                   $x_j^\phi \sim \phi(x)$
      Each possible pairing (x_i^ψ, x_j^ϕ) cast a vote in space of w^3 of weight $\mathrm{wt}(w_i^\psi, x_i^\psi)$
    Keep highest density peak in vote space: w_*^3 of vote score s
    $\mathcal{R} \leftarrow \mathcal{R} \cup (w_*, s)$ with $w_* = w_*^3 \circ w^2 \circ w^1$

Fig. 2: Pose inference algorithm

We also propose an additional step for refining the pose estimate, beyond the precision of the discretized pose values. As illustrated in Fig. 1 (right), we use the peaks identified by the algorithm in the pose space, together with their score value, as approximations of the likelihood function $p(w)$ (Eq. 3) at some discrete "probing" points. We simplistically assume that the main modes in the underlying distribution of W must *locally* approximate a simple isotropic distribution in the pose space. We therefore locally fit such a distribution (isotropic Gaussian and von Mises-Fisher distributions [19]) on the main peaks of $p(w)$, using non-linear least squares. The mean of the fitted distribution is then retained as the peak of that particular mode of the distribution (Fig. 1, right). This provides a much more accurate estimate of the optimal pose(s) compared to the above algorithm (as demonstrated in Section IV-A), at a very small additional computational cost.

C. Weighting of training data

We now present a way of weighting the available training data. The model we use does not include any discriminative

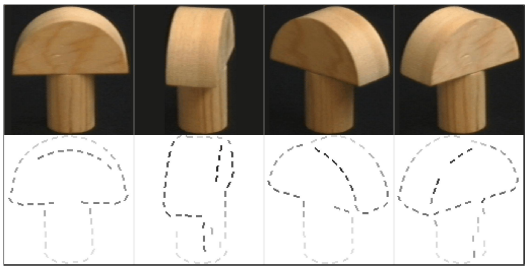


Fig. 3: Visualization of the weights attributed to each image feature (edge fragments) on a toy example; darker colors correspond to heavier weights. The parts looking similar in different views (e.g. the cylindrical base) receive lower weights, while the image features that can unambiguously determine a precise pose (e.g. non-silhouette edges) receive high weights.

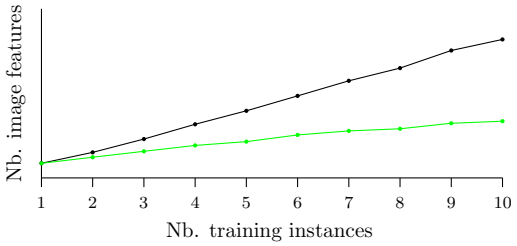


Fig. 4: Size of the category-level model of rotating cars built using different numbers of training instances: without (black) and with (green) the pruning of features by clustering. The proposed approach ensures a sublinear growth of the model.

aspects per se, and this weighting proved to significantly enhance the overall performance of the method (see Section IV). Appropriately weighting training data in the context of object recognition was previously shown to increase performance e.g. in [21], [22], [23], [24]. The formulation proposed here is different, suited to our non-discriminative low-level image features, and does not rely on massive amounts of training examples. The idea is to weight each image feature, depending on how informative it actually is for determining a specific pose. As detailed in the algorithm of Fig. 2, a training feature (w, x) is allowed to cast a vote of weight $\mathrm{wt}(w, x)$, given by

$$\mathrm{wt}(w,x) = 1 - \left[\frac{1}{K} \sum_{w':(w',\cdot)\in\mathcal{T}} \psi'(w',x)\big(1 - \mathsf{K}'_1(w,w')\big)\right] \quad (4)$$

with ψ' and K'_1 being variants of ψ and K_1 with maximum values of 1. This definition yields numerically-convenient weights in the range $[0, 1]$.

In Eq. 4, the expression in square brackets measures, for an image feature x observed in a training pose w, how likely this feature would be in poses very different than w. The weight is then defined using the opposite of that value. This effectively corresponds to the *specificity* of that feature x for the pose w (see also Fig. 3).

III. Learning object *category* models

The model and methods presented above naturally extend to *category*-level models. In that case, the training images include different objects, which together implicitly define the category. This capability of our model is due both to the fact that we can use very simple, non-discriminative image features (points along edges), which often generalize well across different objects of a same category, and by the non-parametric representation of the training data, which can naturally handle variability in the training data, in this case coming from several object instances.

Formally, each object instance $\ell \in [1, L]$ used for training produces a training set $\mathcal{T}_\ell$, as defined in Section II-A. A category-level model is then simply created using all features of all example instances, $\mathcal{T} = \bigcup_\ell \mathcal{T}_\ell$.

A. Pruning of training features by clustering

The above formulation uses a linearly growing number of training points (pose/appearance pairs) as more object instances are used to learn a given category. This correspondingly increases the computational requirements of using the model. Fortunately, object instances within a category often share common appearance traits, and the elements of $\mathcal{T}$ can thus be pruned at a very small cost of the representative capabilities of the model (as shown in Section IV). Practically, the elements of $\mathcal{T}$ are grouped using a simple agglomerative clustering scheme on the joint pose/appearance space, and only the cluster centers are retained. A maximum variance is enforced within the clusters, both in pose and appearance, which determines the amount of discarded training points. Note that the clustering procedure is most efficiently performed after normalizing the training examples from different instances for in-plane translation, rotation and scale, using the hardcoded transformations mentioned in Section II-A.

B. Recognizing a particular trained instance

The clustering of training features limits the size of a category model for efficiency. To compensate for lost accuracy, after identifying an initial pose estimate w_* with this category model, one can determine whether the recognized object corresponds to a specific trained instance. We measure the score of each trained instance ℓ at the pose w_* with

$$p_\ell(w_*) \;=\; \int_{\mathcal{A}} \psi_\ell(w_*, x)\, \phi(x)\, \mathrm{d}x \;, \quad (5)$$

where ψ_ℓ is defined as in Eq. 1, but using only the elements $\mathcal{T}_\ell$ of the instance ℓ. The value is easily approximated [18] with

$$p_\ell(w_*) \;\approx\; \frac{1}{n} \sum_{i}^{n} \psi_\ell(w, x_i) \quad \text{where} \quad x_i \sim \phi(x) \;. \quad (6)$$

If the value of $p_\ell(w_*)$ is significantly higher for a certain ℓ, the corresponding model of that instance ℓ (using all training data available for that instance) is then used to obtain a new, more accurate pose estimate (Section IV-A).

IV. Experimental evaluation

We now evaluate the proposed method under various conditions, using publicly-available datasets. We first analyze the incremental improvements in performance due to the individual ideas proposed in this paper. We then compare our results to existing, competing methods. The image features used are simple points identified along image edges, extracted with the classical Canny detector (see the examples in Fig. 3). Each of those points is characterized by its position in the image, and by the local orientation (smoothed for stability) of the edge at that point (an angle in $[0, \pi[$). As a ballpark figure of efficiency, on a standard laptop, our Matlab implementation of the method takes 20-30 seconds to process an image of the dataset of Section IV-B.

A. COIL Dataset

We first evaluate our method on the classical COIL dataset [25]. This dataset has been used in a variety of contexts, but not in the particular conditions we were interested in. The purpose of this part of our evaluation is to demonstrate the merits of the proposed method, by highlighting the incremental improvements brought by each proposed key point.

We selected a few objects from the original dataset, which correspond to reasonable categories (rectangular boxes, toy cars, flat bottles; see Fig. 5). Most other objects of the dataset were not suitable for estimating their pose (e.g. bell peppers, cylindrical cans) or could not be grouped into categories (e.g. duck toy). The dataset contains 72 images of each object undergoing a full rotation around a single (vertical) axis, with a fixed elevation. The estimated pose is thus similarly limited to this degree of freedom. For training, we use 18 images of each object (thus 20° apart), and the others for testing. We report the error as the median and mean (over all test images) of the absolute error of the estimated orientation. The rectangular boxes and the flat bottles present a 180° rotational symmetry, the error is accordingly evaluated on the half-circle.

1) Seen instances: The first series of tests uses 4 instances of each object (2 for the bottles) for training category models, and those same objects for testing. The basic method (algorithm of Fig. 2 without weighting the training data) already provides accurate results (see Fig. 5), with a median error of 5° which is the best achievable for the nearest-neighbour classification of the algorithm (Fig. 2) iterating on the discrete viewpoint values of the training data. The *mean* error decreases as we use the weights on the training data, as a few ambiguous test images are now better classified, which indicates the superior discriminability between different poses when using those weights. Interestingly, the fitting of a distribution on the pose space over the discrete approximations of the likelihood function (Section II-B) reduces the error significantly, as this allows accuracy beyond the resolution of the nearest-neighbour classification mentioned above. Finally, we refine the pose using the procedure proposed in Section III-B: the pose estimate obtained with the category model is used to efficiently check the resemblance with a particular trained instance. If one trained instance receives a significantly higher likelihood than the others (Eq. 6), its corresponding *instance-specific* model is used to perform a (hopefully) more accurate estimation; this is indeed the case as reported in Fig. 5. This procedure thus makes use of both the category- and instance-models for best efficiency without sacrificing accuracy.

	Toy cars		Boxes		Flat bottles	
Seen instances						
Without weights	5.0	12.1	5.0	13.5	5.0	7.9
With weights on training data	5.0	10.1	5.0	11.6	5.0	8.3
Weights + pruning of train. data	5.0	8.7	5.0	11.0	5.0	6.8
Weights + pruning + fitting of dist.	2.9	7.2	3.4	10.2	5.5	6.1
Refined w/ instance-specific model	2.0	5.8	3.2	9.4	4.2	5.3
Unseen instances						
Without weights	10.0	36.8	10.0	14.4	25.0	28.2
With weights on training data	5.0	39.7	10.0	11.0	30.0	31.2
Weights + pruning of train. data	10.0	44.6	10.0	11.8	15.0	25.5
Weights + pruning + fitting of dist.	2.9	41.8	4.3	8.8	16.8	23.6

Fig. 5: Results of category-level pose estimation with objects from the COIL dataset. Image top row: objects used for training and as *seen* test instances; image bottom row: objects used as *unseen* test instances. We report median (black) and mean (gray) error in degrees; large mean error is caused by (near-)symmetries which often induce errors of 90° and 180°.

2) Unseen instances: The second series of tests uses the same category models, but with a test set of other, *unseen* objects (Fig. 5, second row). The purpose is to verify the generalization capability of the category models. The results, as reported in Fig. 5, show accurate pose estimation results in all of the 3 tested categories, even though the test objects vary in shape, appearance and proportions from the training instances. This is made possible by the combination of different appearance traits of different training instances, which is possible in our non-parametric representation of the model. The flat bottles however yielded slightly worse results, which indicate the difficulty of generalizing the appearance of such objects on the category level. A test view of a novel instance could equally correspond to a wide bottle seen from its side, or to a front-facing thin one.

B. Rotating cars

We evaluated our method using the "Multiview car dataset" used by [7] and [16]. It includes about 2000 images of 20 very different rotating cars filmed at a motor show. The dataset is very challenging due to clutter, changing lighting conditions, high within-class variance in appearance, shape and texture, and highly symmetric side views, or similar front and rear views, which are sometimes hard to discriminate even for a human. The dataset was used in [7] for pose classification in 16 discrete bins, and in [16] for continuous pose estimation.

Number of training examples	15	30	40
Baseline comparison: Torki and Elgammal [16]	5.47	1.93	1.84
Without weights	6.75	3.83	2.94
With proposed weights on training data	6.68	3.81	2.91
Weights + fitting of pose distribution	**4.42**	**1.62**	**1.49**

Fig. 6: Results of pose estimation on a single car; mean error in degrees.

We first evaluated our method, as in [16], on the first car of the dataset, using thus an instance-specific model. We select 15, 30 or 40 equally-spaced images of the sequence as training images, and use all other images (spaced about 3–4° apart) for testing. Using all the key techniques proposed in this paper, we obtain superior results to [16] (see Fig. 6 for details). We then performed an evaluation the "10/10 split", where the first 10 cars of the dataset are used for training, and the other 10 for testing. We obtain again accurate pose estimation results. As highlighted in Fig. 8, most estimated poses are very accurate, while a number have an error of about 180°. This is caused by the symmetric aspects of some cars in the side views, as well as to confusion between front- and rear-facing views. This explains the seemingly large error reported as the mean in Fig. 7, even though the median error is clearly better than the results reported by [16]. In this case, the median as an evaluation metric better reflects the actual precision of the pose estimates, focusing on all the "successful" test cases.

We tested again the generalization capabilities of our model. As proposed in [7], we used the model trained on the cars at the motor show for testing on the database of Savarese *et al.* [4]. The cars appear here in natural environments with more clutter and in very diverse conditions; nevertheless, we obtained interesting results, of which we show some representative examples in Fig. 9. This again demonstrates the good capability of our system to generalize category-level models to conditions very different from those trained for. Note that, unfortunately, no quantitative results for these particular test conditions (proposed in [7]) — that we could compare to — were previously reported.

As a side note, let us mention that we tested our method on this same dataset [4] under the conditions of [8], i.e. training the model with 5 instances of that dataset. We obtained performance on pose estimation of the same order of magnitude as [8], but we missed some information for an exact quantitative comparison (which instances to use for training, and whether or not to include pose estimation results of inaccurate detections). Those experimental conditions were also evaluating coarse pose classification, whereas we focus on continuous pose estimation.

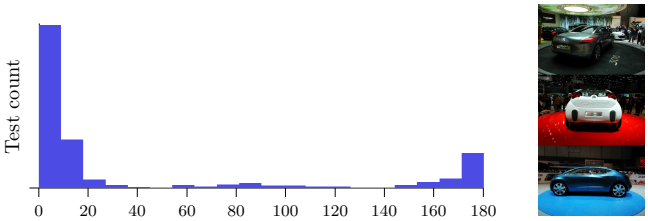


Fig. 8: Histogram showing distribution of error (in degrees) during experiments on multiple cars and sample test images that yielded an error of about 180°, due to ambiguous appearance.

Fig. 9: Detection and pose estimation results on the database of [4], using the model trained with Fig. 7. Boxes indicate the localization of the object as identified by our system, and the roses in the upper-left corners indicate the orientation of the front of the car as seen from the top (as in [7]). The last column contains failure cases, often due to the symmetrical appearance of the cars, or to too much clutter in the background.

V. Conclusions and Future Work

We presented a framework for representing the appearance of object instances or categories, together with its mechanisms to perform object localization and pose estimation in 2D images. The training examples are represented by a probability distribution, stored in a non-parametric manner, in the joint pose/appearance space. This approach can naturally represent a single object, or a whole object category by including different training exemplars of that category. The localization and identification of the pose of the object in a new scene is accomplished via probabilistic voting in the pose space, intrinsically robust to background clutter and occlusions. The overall approach was shown to be competitive or outperform comparable methods. As future work, it will be interesting to evaluate the method in the context of robotic applications,

This article is part of Chapter 5: *Extension from specific objects to object categories* (p.56)
It originally appeared in: *2013 Computer and Robot Vision (CRV)*

	Median	Mean 90%ile	Mean	Error<22.5°	Error<45°	Used training features
Baseline comparison: Ozuysal *et al.* [7]	–	–	46.5	41.7%	71.2%	
Baseline comparison: Torki and Elgammal [16]	11.3	19.4	34.0	70.3%	80.7%	
Without weights on training data	9.3	33.1	47.4	65.1%	70.0%	100%
With weights and fitting of distribution	5.8	23.7	39.0	78.1%	79.7%	100%
Same + moderate pruning of features	6.1	25.8	41.0	77.0%	78.7%	54%
Same + aggressive pruning of features	9.4	32.4	46.8	67.1%	70.0%	30%

Fig. 7: Results of pose estimation on multiple cars; instances 1–10 used for training (top), 11–20 for testing (bottom). Errors of 180° are common (e.g. on instances 16 and 19) and explain the greater mean but smaller median error compared to [16].

with training sets spanning the whole viewing sphere around the objects to learn.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience. Damien Teney is supported by a research fellowship of the Belgian National Fund for Scientific Research.

References

[1] V. Ferrari, T. Tuytelaars, and L. J. V. Gool, "Simultaneous object recognition and segmentation from single or multiple model views," *Int. J. Comp. Vis. (IJCV)*, vol. 67, no. 2, pp. 159–188, 2006. 1

[2] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *Int. J. Comp. Vis. (IJCV)*, vol. 66, no. 3, pp. 231–259, 2006. 1

[3] A. Kushal, C. Schmid, and J. Ponce, "Flexible object models for category-level 3D object recognition," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2007. 1

[4] S. Savarese and L. Fei-Fei, "3D generic object categorization, localization and pose estimation," in *IEEE Int. Conf. on Comp. Vis. (ICCV)*, 2007. 1, 2, 6

[5] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool, "Towards multi-view object class detection," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2006. 1

[6] J. Liebelt, C. Schmid, and K. Schertler, "Viewpoint-independent object class detection using 3D feature maps," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2008. 1

[7] M. Ozuysal, V. Lepetit, and P. Fua, "Pose estimation for category specific multiview object localization," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2009. 1, 2, 5, 6, 7

[8] M. Sun, H. Su, S. Savarese, and L. Fei-Fei, "A multi-view probabilistic model for 3D object classes," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2009. 1, 6

[9] M. Martinez Torres, A. Collet Romea, and S. Srinivasa, "MOPED: A scalable and low latency object recognition and pose estimation system," in *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2010. 1

[10] F. Viksten, R. Soderberg, K. Nordberg, and C. Perwass, "Increasing pose estimation performance using multi-cue integration," in *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2006. 1

[11] C. Gu and X. Ren, "Discriminative mixture-of-templates for viewpoint classification," in *IEEE Europ. Conf. on Comp. Vis. (ECCV)*, 2010. 1

[12] K. Lai, L. Bo, X. Ren, and D. Fox, "A scalable tree-based approach for joint object and pose recognition," in *Conf. on Artificial Intelligence (AAAI)*, 2011. 1

[13] D. Hoiem, C. Rother, and J. M. Winn, "3D LayoutCRF for multi-view object class recognition and segmentation," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2007. 1

[14] P. Yan, S. M. Khan, and M. Shah, "3D model based object class detection in an arbitrary view," in *IEEE Int. Conf. on Comp. Vis. (ICCV)*, 2007. 1

[15] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and continuous pose estimation," *Image and Vision Computing*, 2012. 1, 3

[16] M. Torki and A. M. Elgammal, "Regression from local features for viewpoint and pose estimation," in *IEEE Int. Conf. on Comp. Vis. (ICCV)*, 2011. 2, 5, 6, 7

[17] A. Opelt, A. Pinz, and A. Zisserman, "Learning an alphabet of shape and appearance for multi-class object detection," *Int. J. Comp. Vis. (IJCV)*, 2008. 2

[18] D. Teney and J. Piater, "Generalized Exemplar-Based Full Pose Estimation from 2D Images without Correspondences," in *Digital Image Computing: Techniques and Applications (DICTA)*, 2012. 2, 3, 4

[19] R. Detry and J. Piater, "Continuous surface-point distributions for 3D object pose estimation and recognition," in *Asian Conf. on Comp. Vis. (ACCV)*, 2010. 3

[20] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comp. Vision (IJCV)*, vol. 77, no. 1-3, pp. 259–289, May 2008. 3

[21] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," in *IEEE Int. Conf. on Comp. Vis. (ICCV)*, 2007. 4

[22] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2009. 4

[23] S. Maji and J. Malik, "Object detection using a max-margin hough transform," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2009. 4

[24] P. Yarlagadda and B. Ommer, "From meaningful contours to discriminative object shape," in *IEEE Europ. Conf. on Comp. Vis. (ECCV)*, 2012. 4

[25] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library COIL-100," Columbia University, Tech. Rep., 1996. 5

Chapter 6

Modeling and using changes of appearance between discrete viewpoints

The two previous chapters use the "exemplar-based" approach of multiview models, which describe the appearance of an object at multiple discrete viewpoints. We however argue that this paradigm is not really satisfactory, because it models each viewpoint independently without taking into account the relations between those viewpoints in terms of appearance and similarity. The contributions of this chapter consist in the explicit identification and representation of the changes of appearance between viewpoints, and in the use of this additional information, stored in the model, to help two different tasks, described below.

The existing approaches to identify changes of appearance with respect to continuous variations of the viewpoint usually consist either in the tracking of image features [57] (e.g. interest points) in videos of the object, or in the matching of precise landmarks between views using local descriptors [58] (e.g. SIFT features). We are however also interested in models made of dense and thus potentially unmatchable features. We therefore found more appropriate to detect dense deformations between adjacent viewpoints, by estimating the optical flow between pairs of images (of two adjacent viewpoints). This results in the estimation, for each pixel in one image, of its translation to another location in the other image. These translations can be applied to any image feature extracted from one image, and the translation can be interpolated between those two images to estimate the appearance at an intermediate viewpoint.

First, we use this explicit modeling of deformations to perform continuous pose estimation in a novel image. Assuming the object of interest has been identified (localized) in the image at one of the learned viewpoints, we then use the generative model to optimize for a better, more accurate viewpoint. With a simple hill-climbing optimization algorithm, we iterate between the generation of the appearance of the object at a slightly perturbed viewpoint, and the measure of its similarity with the test image, until convergence to a local maximum. Note that we initially apply this procedure to

specific objects only, in order to keep the discussion simple. We will apply it to object categories in Chapter 8.

Second, we use the modeling of deformations for pose estimation in a context of active vision, in order to disambiguate poses of similar appearance. Similarly to the rest of our work, only a single image of the test scene is initially available. The robotic agent is however then free to observe the scene under additional, chosen viewpoints, to help resolve ambiguities. Such ambiguities are especially frequent with objects presenting little texture or few internal edges. The appearance of such an object is then mainly determined by its silhouette, which appears almost identical as the object is pointing *towards* or *away* from the camera. In this case, the robot would take a second picture of the scene, after a slight displacement relative to the first one. As above, we then use optical flow to identify the parallax between the two images. The key is then to compare this information with the one stored in the model: the parallax will be in opposite directions whether the object was pointing towards or away from the camera, which will thus resolve this ambiguity.

The paper included in the following pages was presented at the 2013 *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*.

Modeling Pose/Appearance Relations for Improved Object Localization and Pose Estimation in 2D images

Damien Teney
University of Liège, Belgium
Damien.Teney@ULg.ac.be

Justus Piater
University of Innsbruck, Austria
Justus.Piater@UIBK.ac.at

Abstract. We propose a multiview model of appearance of objects that explicitly represents their variations of appearance with respect to their 3D pose. This results in a probabilistic, generative model capable of precisely synthesizing novel views of the learned object in arbitrary poses, not limited to the discrete set of trained viewpoints. We show how to use this model on the task of localization and full pose estimation in 2D images, which benefits from its particular capabilities in two ways. First, the generative model is used to improve the precision of the pose estimate much beyond nearest-neighbour matching with training views. Second, the pose/appearance relations stored within the model are used to resolve ambiguous test cases (e.g. an object facing towards/away from the camera). Here, changes of appearance as a function of incremental pose changes are detected in the test scene, using a pair or triple of views, and are then matched with those stored in the model. We demonstrate the effectiveness of this method on several datasets of very different nature, and show results superior to state-of-the-art methods in terms of accuracy. The pose estimation of textureless objects in cluttered scenes also benefits from the proposed contributions.

1 Introduction and related work

We focus on the problem of 3D pose estimation of known objects in 2D images, using multiple registered images of the objects as training examples. Pose estimation, which is closely coupled to the related tasks of object recognition and localization, is a fundamental problem in computer vision and has naturally received great interest over the years. The main contribution of this paper is to explicitly include, in an existing multiview model of appearance [14], the possible changes of appearance undergone by the object as it pose varies between the trained viewpoints. With the exception of [9], this is, to our knowledge, the only work to include such information within a model of appearance in the context of pose estimation. We make use of this additional information in two different ways to improve the precision and accuracy of pose estimation. In the following we relate our approach to related work.

Multiview models of appearance. The traditional methods for object recognition using 2D images alone, known as appearance-based, typically use

This article is part of Chapter 6: *Modeling and using changes of appearance between discrete viewpoints* (p.61)
It originally appeared in: *2013 Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*

specific models for individual viewpoints, e.g. a model for cars seen from the front, and another for cars seen from the side. Recent contributions in object recognition have introduced more and more models of appearance that include different viewpoint and that are also relevant to pose estimation. Some methods still treat those different viewpoints somewhat independently [14,18], while others try to match and link features across viewpoints [5,10,16]. Savarese *et al.* [10], for example, model an object as a collection of planar parts that can appear in different views. We follow an intermediate approach, by storing independently the image features that make up the different views, but we also store, along with every each image feature, how its appearance varies with respect to the pose of the object. The multiview models mentioned above were mainly used on the task of localization and recognition, without recovering a 3D pose explicitly, or only as rough estimate such as "frontal view" or "side view". We rather focus on *continuous* pose estimation, to recover precise 3D position and orientation, as is needed, e.g. for robotic applications [7,18].

Continuous pose estimation. The classical approach to pose estimation using 2D training examples is to match highly discriminative features between the test and training views. These matches then vote for the most similar training example, yielding a nearest-neighbour classification of limited precision. Some authors have proposed averaging [18] and probabilistic smoothing [14,15] schemes to increase precision beyond the resolution of the training examples on the viewing sphere. While these procedures basically perform some averaging between trained viewpoints, we rather explicitly detect, and include in the model, the deformations and the transformations of appearance between the discrete viewpoints seen during training. We then use this information in our generative model to finely optimize the 3D pose, starting from a rough nearest-neighbour estimate. Another, radically different approach was proposed by Torki and Elgammal [17], who learn a regression from local image features to the pose of the object. This original approach recovers a precise pose, but cannot handle significant clutter or occlusions, and the accurate pose estimation depends on the (supervised) enforcement of a one-dimensional manifold constraint (corresponding to the 1D rotation of the object in the training examples). It is not clear how that approach would extend to the estimation of the full 3D pose of an object.

New view synthesis. Our approach uses dense optical flow to identify the deformations between pairs of neighbouring training views. Only those parts of this dense information are then retained that correspond to the sparse image features actually stored in the model. This information can then be used in a generative manner, to synthesize the appearance of the object in a new, unseen pose, by transforming the image features of nearby trained viewpoints according to those stored deformations. The problem of new view synthesis has been studied in the field of computer graphics through the technique of *morphing* [1,2,11]. Most methods only consider pairs or triples of views, whereas we are interested in modeling and using transformations over the whole viewing sphere. Morphing algorithms also often rely on established correspondences between specific

This article is part of Chapter 6: *Modeling and using changes of appearance between discrete viewpoints* (p.62)
It originally appeared in: *2013 Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*

image features of the input views [11], whereas we use dense optical flow to identify deformations between neighbouring views, before applying them to sparse features. As an advantage, our approach readily applies to difficult-to-match features (as opposed to the competing method of Savarese *et al.* [9]). This practically allows handling non-textured objects containing little detail. Although some global consistency in the detected deformations is enforced by optical flow algorithm, each image feature independently stores its possible deformations. This does not limit the model to a particular class of overall transformations. On the contrary, Savarese *et al.* [9] specifically models affine transformations of object parts, assuming that large planar parts can be identified (which is not a universal property of objects). Note also that we use a sparse set of training views (typically spaced about 20° apart on the viewing sphere) and do not require videos or dense sequences of images to track features between frames, as opposed to Sun *et al.* [13]. One may also note a similarity in spirit with the classical active appearance models used mainly for object tracking; they are however based on and limited by point-wise matches of specific landmarks.

Active vision. In addition to the generative model we use to refine pose estimates, we show how to use the deformations stored in the model to resolve ambiguous test cases (Section 4). In such scenes, the 2D appearance of the object can equally correspond to several 3D poses (see Fig. 4 for an example). We propose to also identify the changes of appearance with respect to the pose in the *test scene*. The camera is therefore allowed to move slightly in two orthogonal directions on the viewing sphere (around the test scene). The changes of appearance are detected — as with the training views — and used as extra dimensions in the descriptor of the image features. The features of the test scene can then be matched more discriminatively with those of the training data, and effectively identify the single correct pose unambiguously. This procedure, which can prove crucial in robotic applications, has been proposed in the field of active vision [3,12], but not integrated, to our knowledge, in such a straightforward manner within a pose estimation method. It resembles the way humans themselves resolve such perception ambiguities (in addition to stereo vision) by moving around the scene. Note that a similar effect can be obtained by fusing the result of independent pose estimations from multiple 2D images [14,18]. Our integrated procedure is however arguably more efficient, the displacement (change of camera position) does not need to be precisely know, and can also be very small (theoretically infinitesimal small, even though image noise and resolution dictate minimum displacements in practice).

2 Representation of training and test views

2.1 Notations for image features and object poses

The contributions of this paper are integrated with the method proposed in earlier work [14]. That method performs object recognition and pose estimation in 2D images, and is applicable to various types of image features. This includes features that cannot easily be matched between training and test views, such as

This article is part of Chapter 6: *Modeling and using changes of appearance between discrete viewpoints* (p.63)
It originally appeared in: *2013 Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*

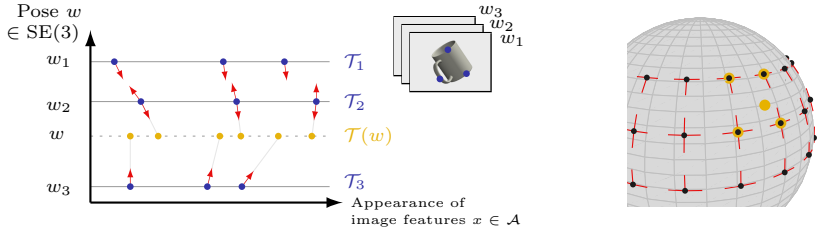


Fig. 1: Left: schematic representation of training data, in the dimensions of pose and appearance. Image features (blue points) are extracted from training images, and their possible changes of appearance (red arrows) are identified between neighbouring views. The appearance of the model at a novel view w (orange) is generated by adjusting the features of close-by views, according to those stored deformations. Right: representation of a set of training viewpoints (black dots) on the viewing sphere. The changes of appearance are detected between pairs of neighbouring views (red links). The appearance of a novel view (orange dot) is generated using the closest training viewpoints, four in this case (orange circles).

the edge points we use in our implementation. We will first review the notation for representing the test and training views, followed by the generative model, capable of synthesizing novel views.

The test data corresponds to a single 2D image of a scene, from which we extract image features. In general, an image feature x is defined by its localization in the image x^{p}, and an optional appearance descriptor x^{a}. Together, they are defined on the *appearance* space $\mathcal{A}$. Practically, we use points identified along edges, combined with the local orientation of the edge (see Fig. 3), so that $\mathcal{A} = \mathbb{R}^2 \times S_1^+$ (the 2D localization plus an orientation without direction). The image features from the test view form a set of *observations* $\mathcal{O} = \{x_i\}_i$, with $x_i \in \mathcal{A}$. The training data correspond to a series of K images of the object of interest, in different poses $w_k \in \mathrm{SE}(3)$ with $k = 1, \ldots, K$. Image features are extracted from each training view k to form a set $\mathcal{T}_k = \{x_i\}_{i=1}^{M_k}$, with $x_i \in \mathcal{A}$.

A pose $w \in \mathrm{SE}(3)$, which defines a 3D location together with a 3D orientation, conveniently decomposes into separate sets of dimensions w^{v} and w^{t}. We call w^{v} the *viewpoint transformations* (defining which side of the object is facing the camera, i.e. an element of S^2) and w^{t} the set of *in-plane transformations* (i.e. translations and rotations parallel to the image plane, and depth/scale changes). In-plane and viewpoint transformations are considered separately, since the changes of appearance induced by the former are fixed by the calibration of the camera. The calibration is assumed to be known, and those transformations can thus be formally hard-coded in the function $\mathrm{transformInPlane}_{w^{\mathsf{t}}}(\mathcal{T}) = \mathcal{T}'$, which transforms a set of image features $\mathcal{T}$ according to the in-plane transformations w^{t}. Without loss of generality, the following discussion will assume that the training views have been normalized for in-plane transformations, that is, centered and set to a similar scale/rotation[1].

[1] Formally, $\mathcal{T}_k \leftarrow \mathrm{transformInPlane}_{-w_k^{\mathsf{t}}}(\mathcal{T}_k)$ and $w_k^{\mathsf{t}} \leftarrow 0, \ \forall k$.

2.2 Generative model of training data

The training data, as presented above, defines the appearance of the object of interest at a set of discrete trained viewpoints. The goal of our generative model is to fill in the gaps between those viewpoints. Although it may be possible to establish explicit correspondences between image features of nearby training images, this approach may not always be reliable, and it does not generalize to dense or non-discriminative image features such as our edge points. Therefore, we choose to identify *dense* deformations between pairs of adjacent training views, using an optical flow algorithm. Those deformations will then be combined to deform the image features of the training images into the novel viewpoint.

More precisely, for an arbitrary viewpoint w^{v}, we identify its closest training viewpoints $\mathrm{nb}(w^{\mathsf{v}}) = \{k : d(w^{\mathsf{v}}, w_k^{\mathsf{v}}) \leq t\}$, where $d(\cdot,\cdot)$ measures the angular distance between two viewpoints. The threshold t is chosen similar to the typical angular distance between neighbouring viewpoints in the training data. We also identify the set of all neighbouring training viewpoints as $\mathrm{NB} = \{(k,k') : k' \in \mathrm{nb}(w_k^{\mathsf{v}}), k \neq k'\}$. During an off-line training phase, an optical flow algorithm [6] is applied on all pairs of views $(k,k') \in \mathrm{NB}$. Each pair produces a dense flow map $\mathrm{UV}_{k\to k'}(x)$ that corresponds, in our case, to the local deformation (translation in the image plane) undergone at an image location x when moving from viewpoint k to k'. Although we compute a *dense* optical flow, we only need to store the actual values of the maps UV for the positions of the few image features of each view. We can now define our generative model $\mathcal{T}(w)$, which produces a set of image features corresponding to the appearance of the object in an arbitrary pose w. Its definition combines the image features of all nearby training views, individually translated using the stored deformations, then adjusted for in-plane transformations. Formally,

$$\mathcal{T}(w) \;=\; \bigcup_{k\in\mathrm{nb}(w^{\mathsf{v}})} \mathrm{transformInPlane}_{w_k^{\mathsf{t}}\to w^{\mathsf{t}}}\left(\; \mathrm{deform}_{w_k^{\mathsf{v}}\to w^{\mathsf{v}}}(\mathcal{T}_k)\;\right). \qquad (1)$$

The functions transformInPlane() and deform() adjust a set of image features respectively for in-plane and out-of-plane transformations. While the definition of the former is trivial (it just applies the translation, scaling and rotation of its parameter), the latter is more complex. It uses a linear combination of two available deformations to translate each image feature. We denote those two deformations by the indices of the viewpoints that generated them, and call them (k,k') and (k,k''). They are chosen from NB so that the novel viewpoint can be reached (on the viewing sphere) by a positive linear combination of them. Consequently, $\exists\, \alpha,\beta \in \mathrm{R}^+ : w^{\mathsf{v}} = w_k^{\mathsf{v}} + \alpha(w_{k'}^{\mathsf{v}} - w_k^{\mathsf{v}}) + \beta(w_{k''}^{\mathsf{v}} - w_k^{\mathsf{v}})$. Practically, this means that the viewpoints k, k' and k'' cannot be collinear on the viewing sphere. With training viewpoints spaced on a grid, as in our experiments, we simply choose k' and k'' respectively along the changes in azimuth and elevation. It is now straightforward to define the deform() function that combines those two chosen deformations:

$$\begin{aligned}\mathrm{deform}_{w_k^{\mathsf{v}}\to w^{\mathsf{v}}}(\mathcal{T}_k) = \{x_i' :\quad & x_i'^{\mathsf{p}} = x_i^{\mathsf{p}} + \alpha\,\mathrm{UV}_{k\to k'}(x_i^{\mathsf{p}}) + \beta\,\mathrm{UV}_{k\to k''}(x_i^{\mathsf{p}}) \\ \text{and}\quad & x_i'^{\mathsf{a}} = x_i^{\mathsf{a}}\ ,\quad \forall x_i \in \mathcal{T}_k\ \}\,.\end{aligned} \qquad (2)$$

3 Refinement of pose with generative model

3.1 Method

The proposed generative model readily integrates with the method proposed in [15]. That method for pose estimation relies on continuous distributions of image features in the appearance space to represent the training and test views, using kernel density estimation. These distributions are simply built using the elements of $\mathcal{O}$ and $\mathcal{T}(w)$ as particles, giving respectively $\phi_{\mathcal{O}}(x) = \frac{1}{|\mathcal{O}|} \sum_{x_i \in \mathcal{O}} \mathsf{K}(x, x_i)$ and $\phi_{\mathcal{T}(w)}(x) = \frac{1}{|\mathcal{T}(w)|} \sum_{x_i \in \mathcal{T}(w)} \mathsf{K}(x, x_i)$, with $\mathsf{K}(\cdot, \cdot)$ a kernel on $\mathcal{A}$.

We reuse the base method proposed in [15] to obtain initial proposals for the 3D pose of the object in the new scene. That method iterates over the training viewpoints and some discrete values of scale and in-plane rotation, then uses a probabilistic voting scheme between matching training and test features to identify the most probable image location. The peaks with high voting scores are then retained as initial pose estimates. This method basically corresponds to a nearest-neighbour identification of the training views in the test scene, and gives us initial estimates to refine by local optimization.

Using the two distributions of image features presented above, and a cross-correlation between them as a measure of similarity [14,15], we now have a likelihood function that can be used to evaluate any arbitrary pose w:

$$\begin{aligned} p(w) \;&=\; \int_{\mathcal{A}} \phi_{\mathcal{O}}(x)\, \phi_{\mathcal{T}(w)}(x)\, \mathrm{d}x \;, \quad \text{approximated with} \\ p(w) \;&\approx\; \frac{1}{n} \sum_{i}^{n} \phi_{\mathcal{O}}(x_i) \quad \text{where} \;\; x_i \sim \phi_{\mathcal{T}(w)} \;, \end{aligned} \tag{3}$$

using Monte Carlo integration, which gives a convenient expression, relatively inexpensive to evaluate. This function $p(w)$ constitutes the objective function that we seek to maximize when optimizing a pose estimate. In practice, it is generally smooth in the neighbourhood of the global optimum, but no assumption can be made about its convexity, and its definition on the 6-dimensional pose space SE(3) makes the evaluation of its gradient difficult. Fortunately, our initial pose estimate can be assumed to be a close approximation of the global optimum. All those conditions motivated the use of a simple hill-climbing algorithm. We iteratively optimize pairs of dimensions at a time, namely the 2 viewpoint angles, the image location, then the scale and in-plane rotation. We empirically observed that a close approximation of the global optimum can be reached in this way after only a few iterations (see Fig. 5, bottom right).

3.2 Results

Rotating car. We first evaluate our method on the first sequence of the "rotating car" dataset [8]. It consists of 118 images of a car on a rotating platform, shot at a motor show. Although it only includes a single degree of freedom (the rotation around the vertical axis), this dataset is interesting as it was shot in real

This article is part of Chapter 6: *Modeling and using changes of appearance between discrete viewpoints* (p.66)
It originally appeared in: *2013 Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*

conditions, features a highly textured object of complex structure and allows a comparison of precision with two state-of-the-art methods. We report in Fig. 2 the precision (error of the estimated rotation angle of the car) of our initial pose estimate and of the pose estimates optimized using our generative model (Section 3.1). We show a clear advantage, with different sizes of training sets (which contain uniformly spaced images from the sequence).

Number of training views	**15**	**30**	**40**
Baseline comparison 1 [17]	5.47	1.93	1.84
Baseline comparison 2 [15]	4.42	1.62	1.49
Initial estimate (nearest neighbour)	6.79	3.78	3.00
Optimized w/ generative model	**1.99**	**0.92**	**0.78**

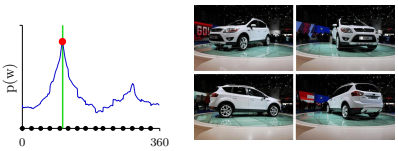


Fig. 2: Results of pose estimation on the rotating car dataset; mean error in degrees. Center: for one test image, we verify that our objective function (blue; evaluated over the whole range of the 1D rotation for demonstration) presents its global optimum near the ground truth (green line). Also represented: training poses (black dots) and our optimized result (red dot). Right: samples test images.

3D pose dataset. We now evaluate our method using the "3D pose dataset" of Viksten *et al.* [4,19]. It is one of the few public datasets available with views spanning more than a 1D rotation, and precisely annotated (in this case with the azimuth/elevation angles of the viewpoint). We use the only object (Volvo car) that was evaluated individually [4], using the same experimental conditions. This allows a comparison with a classical method [4], which uses discriminative image descriptors with a voting and averaging scheme; this constitutes the classical approach for robust 3D pose estimation. The small and large training sets contain views spaced respectively 20° and 10° apart (on both azimuth and elevation angles), with test views in between. With the larger training set, we obtain results superior to [4] in terms of accuracy (Fig. 3). The smaller training set is more challenging for detecting deformations between views, reaching the limits of the optical flow algorithm we use. Our large mean error is caused by two views that yield wrong initial pose estimates, with a large error of almost 180°, due to the similar aspect of the front and rear of the car.

Spacing between train. views	**20°**	**10°**
Baseline comparison [4]	4.21° / 2.66°	1.25° / 1.06°
Initial estimate (near. neigh.)	34.44° / 10.00°	5.14° / 5.00°
	5.00° / 5.00°	1.23° / 1.23°
Optimized w/ gen. model	27.22° / **1.78°**	**0.92°** / **0.88°**
	2.00° / 2.00°	1.00° / 1.00°

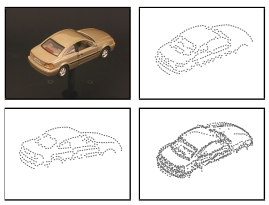


Fig. 3: Results on 3D pose dataset; mean (median) error of azimuth/elevation angles. Clockwise: one test image, its image features, features produced by our generative model at the optimized pose, and features of closest training view; the generative model closely approximates the appearance of the unseen view.

This article is part of Chapter 6: *Modeling and using changes of appearance between discrete viewpoints* (p.67)
It originally appeared in: *2013 Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*

4 Matching pose/appearance relations in ambiguous test scenes

4.1 Method

We propose to make use of the pose/appearance relations identified and stored within the model in a second manner, as extra dimensions of the descriptor of the image features. In this context, the same changes of appearance with respect to the pose are identified in the test view, using additional images obtained by moving the camera slightly around the test scene (which effectively changes the relative pose between the camera and the object of interest). Those additional images are only used to identify the deformations (as in Section 2.2); only the features of the original test image are actually used. Each image feature $x \in \mathcal{A}$ of both the training set and the test image is then complemented with an extra information x^{d}, a first-order approximation of the derivative of its position in the image with respect to the viewpoint, i.e. $x^{\mathsf{d}} \approx \frac{\partial x^{\mathsf{p}}}{\partial w^{\mathsf{v}}}$ ($\in \mathcal{R}^2 \times \mathcal{R}^2$). This conveniently constitutes a compact representation of the local deformations. In practice, considering an image feature x of a training view k, we approximate x^{d} by averaging the deformations identified with the neighbouring views:

$$x^{\mathsf{d}} \;=\; \text{average}_{k' \in \text{nb}(k)} \left(\frac{\text{UV}_{k \to k'}(x^{\mathsf{p}})}{w^{\mathsf{v}}_{k'} - w^{\mathsf{v}}_{k}} \right) . \tag{4}$$

Using azimuth and elevation angles to parametrize a viewpoint on the 2-sphere, this expression gives us two vectors (each $\in \mathcal{R}^2$), corresponding to the translation in the image place relative resp. to azimuth and elevation changes of the viewpoint. These extra feature descriptors are similarly extracted in the test view. We then use them, both when matching observations between the training and test views for voting for an initial pose estimate, and when measuring the similarity between the test view and a generated view (Section 3.1). In both cases, we set a hard threshold for classifying two features x_1 and x_2 as similar[2]:

$$\text{angle}(x_1^{\mathsf{d}}, x_2^{\mathsf{d}}) < 135^\circ \quad \textit{or} \quad \|x_1^{\mathsf{d}}\| < t' \quad \textit{or} \quad \|x_2^{\mathsf{d}}\| < t' \; . \tag{5}$$

The threshold t' on the magnitude of the deformations discards small and insignificant deformations, which cannot be identified reliably. The function $\text{angle}(\cdot, \cdot)$ measures the difference in direction between the two deformations. The maximum value of 135° discards matches of *truly opposite* directions (as is the case with the ambiguous situations we are interested in), while still keeping most (even uncertain) matches (maybe simply due to noise), which is important in the voting algorithm [15] for the initial pose estimate.

4.2 Results

We finally evaluate the second proposed use of image deformations, for resolving ambiguous test cases by matching them. No dataset suitable for this very

[2] To shorten notations, we express the condition on a single vector, but it must be verified by both parts of x^{d}.

This article is part of Chapter 6: *Modeling and using changes of appearance between discrete viewpoints* (p.68)
It originally appeared in: *2013 Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*

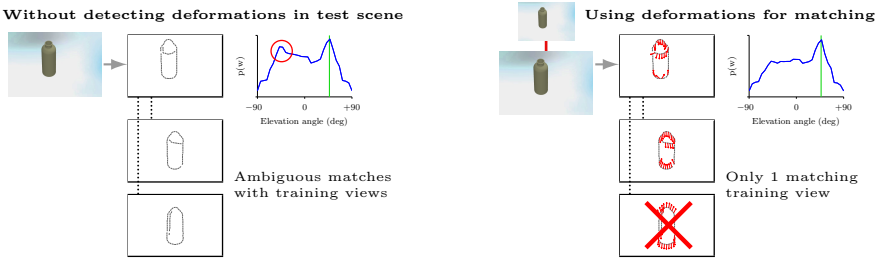


Fig. 4: Left: ambiguous test scene. The extracted edges correspond equally well to training images of the bottle facing towards/away from the camera, and our pose likelihood function $p(w)$ presents two strong peaks (incorrect one in red, ground truth in green). Right: using a second image taken after moving the camera slightly to the top, we detect deformations of image features (red arrows), and match them with the training examples; only the correct peak of $p(w)$ remains.

Objects					
No deform. + near. neigh.	55% 6.3°	80% 5.0°	68% 6.5°	69% 5.2°	51% 16.7°
Match. def. + near. neigh.	66% 6.2°	82% 16.6°	81% 6.2°	79% 5.4°	60% 17.5°
Match. def. + gen. model	70% 5.0°	82% 14.0°	92% 4.6°	77% 1.4°	60% 14.7°

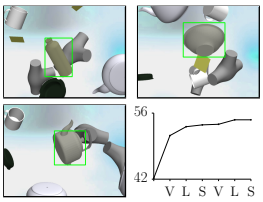


Fig. 5: Left: results on synthetic scenes without/with matching of deformations; success rate (localization error < 20px and pose error $< 20°$) and mean pose error of successes. Right: sample test images with localization results as bounding boxes. Bottom right: typical evolution of our objective function after successive optimizations for viewpoint, image localization and scale/in-plane rotations.

particular problem is currently available, and we resorted to synthetic images, featuring simple objects. Although simple in appearance, they actually prove challenging for pose estimation due to their lack of detail and texture. The test data is now a "central" 2D image, complemented by 2 additional views obtained by moving the camera slightly to the right and to the top. This allows recovering the changes of appearance of the scene with respect to the pose, and matching them with the training data (Fig. 4). As reported in Fig. 5, this eases the localization and improves the precision of the pose estimation.

5 Conclusions

We integrated, within a multiview model of appearance, the explicit transformations undergone by the image features between the training viewpoints. The deformations between example images are detected with dense optical flow and stored for the discrete image features. First, we used this information in a generative model, to refine an initial estimate of the 3D pose of the object in a

This article is part of Chapter 6: *Modeling and using changes of appearance between discrete viewpoints* (p.69)
It originally appeared in: *2013 Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*

new scene. Second, we showed how to match deformations between training and test data, in order to resolve the pose in ambiguous test images. We clearly demonstrated the advantage of those contributions on several datasets, and over existing methods. As future work, it will be interesting to integrate and evaluate these principles within the context and practical conditions of robotic applications.

Acknowledgments The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement no. 270273, Xperience. Damien Teney is supported by a research fellowship of the Belgian National Fund for Scientific Research.

References

1. Avidan, S., Shashua, A.: Novel view synthesis in tensor space. In: CVPR (1997) II
2. Chen, S.E., Williams, L.: View interpolation for image synthesis. In: SIGGRAPH (1993) II
3. Chen, S., Li, Y., Kwok, N.M.: Active vision in robotic systems: A survey of recent developments. Int. J. of Rob. Res. 30(11), 1343–1377 (2011) III
4. Johansson, B., Moe, A.: Patch-duplets for object recognition and pose estimation. In: CRV (2005) VII
5. Kushal, A., Schmid, C., Ponce, J.: Flexible object models for category-level 3D object recognition. In: CVPR (2007) II
6. Liu, C.: Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. Ph.D. thesis, MIT (2009) V
7. Martinez Torres, M., Collet Romea, A., Srinivasa, S.: MOPED: A scalable and low latency object recognition and pose estimation system. In: ICRA (2010) II
8. Ozuysal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: CVPR (2009) VI
9. Savarese, S., Fei-Fei, L.: View synthesis for recognizing unseen poses of object classes. In: ECCV. Marseille, France (2008) I, III
10. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: IEEE Int. Conf. on Comp. Vis. (2007) II
11. Seitz, S.M., Dyer, C.R.: Toward image-based scene representation using view morphing. In: Int. Conf. on Patt. Rec. pp. 84–89 (1996) II, III
12. Sipe, M.A., Casasent, D.: Best viewpoints for active vision classification and pose estimation. In: Intelligent Robots and Comp. Vis. pp. 382–393 (1997) III
13. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3D object classes. In: CVPR (2009) III
14. Teney, D., Piater, J.: Generalized Exemplar-Based Full Pose Estimation from 2D Images without Correspondences. In: DICTA (2012) I, II, III, VI
15. Teney, D., Piater, J.: Continuous pose estimation in 2D images at instance and category levels. Submitted (2013) II, VI, VII, VIII
16. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiel, B., Van Gool, L.: Towards multi-view object class detection. In: CVPR (2006) II
17. Torki, M., Elgammal, A.M.: Regression from local features for viewpoint and pose estimation. In: ICCV (2011) II, VII
18. Vikstén, F., Soderberg, R., Nordberg, K., Perwass, C.: Increasing pose estimation performance using multi-cue integration. In: ICRA (2006) II, III
19. Vikstén, F., Forssén, P.E., Johansson, B., Moe, A.: Comparison of local image descriptors for full 6 degree-of-freedom pose estimation. In: ICRA (2009) VII

Chapter 7

Use of intensity gradients as dense image features

The model of object appearance of Chapter 5 was initially presented in a general formulation with respect to image features, but only applied so far to edge segments. The present chapter considers the use of intensity gradients, extracted at a coarse scale and densely defined across entire images. Using this type of image information was one of our initial aims, since such gradients are particularly helpful to represent shape properties, by capturing shading onto smooth surfaces. Using gradients as image features is thus particularly useful to recognize and determine the pose of objects of smooth shapes, with little texture or few discriminative visual characteristics.

In this chapter, for the clarity of the discussion and in order to keep the focus on the image features, we do not discuss the task of pose estimation explicitly. The main contribution is to adapt the representation of appearance, so as to accommodate the dense image gradients, through the definition of suitable kernel functions. The variety of tasks considered in the evaluation section may appear surprising. This is a consequence of using a common formulation for all these tasks, simply through the use of our measure of visual similarity between the model and the test image.

The paper included in the following pages has not been published and is still to be submitted.

Probabilistic Templates of Dense Image Features for Object Detection and Recognition

Damien Teney
damien.teney@ulg.ac.be
University of Liège
Belgium

Justus Piater
justus.piater@uibk.ac.at
University of Innsbruck
Austria

Abstract

This paper presents a general formulation for modeling the 2D appearance of objects and performing recognition in images, which bridges the gap between the traditional feature- and appearance-based approaches. The proposed model represents appearance as probability distributions in the image space, defined in a non-parametric manner using sparse or *dense* image features from training examples. Its ability to handle dense image information — in addition to classical sparse features such as interest points or edges — is demonstrated using image gradients, defined at pixel resolution. Those gradients, e.g. due to shading on smooth surfaces, can help resolve cases where the appearance of edges alone proves equivocal. We thus present a common probabilistic framework for handling very different types of image information. We define a measure of similarity using cross-correlation of distributions, and perform detection of objects in cluttered images by identifying the local maxima of this similarity between the test image and the learned model. This is performed efficiently via a voting procedure, reminiscent of the classical Hough scheme. Probabilistic techniques such as Monte Carlo integration and importance sampling are used to ensure balance between accuracy and efficiency. Finally, the non-parametric representation readily accounts for variability in appearance of different training exemplars, due, for example, to different conditions of illumination observed during training when using image gradients. Beyond its technical flexibility and the theoretical rigor of its formulation, the proposed method proves competitive on existing datasets, where we demonstrate consistent and significant improvement of recognition performance by using the proposed gradient features over traditional edges alone.

1 Introduction and related work

As one of the main tasks in the field of computer vision, the recognition of objects in 2D images has received lots of attention over the years. The way of performing recognition depends on the internal representation chosen to model the appearance of the objects. While some methods seek to build an intrinsic, geometrical 3D model of the object [9], we focus on the 2D appearance alone, and train our model with simple example images. Another level of distinction among recognition methods is the focus either on specific objects or on object categories. The proposed method handles variability in appearance in a probabilistic way; this variability among the training examples can equally come from different objects of a same category, or from variations of appearance of a unique object, e.g. observed under

This article is part of Chapter 7: *Use of intensity gradients as dense image features* (p.72)
It has not been published and is still to be submitted.

different conditions of illuminations, as will be used with the proposed gradients as image features.

Some classical methods for object recognition make use of the appearance of the object as a whole [3, 5, 7]. Among those so-called *appearance-based* methods, a common trait is a poor robustness to occlusions, together with the need of large numbers of training views. At the opposite end, *feature-based* methods rely on matching characteristic, local features between the test image and the training examples [9]. This relies on the extraction of discriminant image features, such as interest points with SIFT descriptors, which often fails with non-textured objects.

Most current, state-of-the-art methods for object recognition rely on the use of image edges (e.g. [6], among many others). The typical technique basically consists in building intermediate representations such as contour fragments, which can then be matched discriminatively between training and test images, and used e.g. in a Hough voting scheme to perform detection in clutter. The approach proposed in this paper, by contrast, leverages the simplicity of low-level, fine-grained image features, which we do not seek to match between training and test images. This allows us to use indiscriminate and *dense* image features such as gradients. Recognition and detection of an object in clutter is based on a similarity measure that uses the cross-correlation of the distributions of features in the training and test images. The advantage is the generality of the formulation, suitable to various kinds of image features, provided such distributions are properly defined. Note that, in the case of edges, correlation-based distance measures have been proposed before. Most notably, the successful directional chamfer distance [4, 11] can be seen as an approximation of our formulation.

The key advantage offered by the proposed system is its ability (1) to effectively handle non-textured objects, and, even more importantly, (2) to resolve cases where edges alone only offer ambiguous information on the presence or the pose of an object in a scene. This allows us to significantly improve detection performance in clutter, as we do demonstrate on existing datasets. It also allows solving cases where shading is the sole relevant clue, e.g. to determine whether a can (cylinder) is viewed from its hollow or flat end (see Section 4.1).

Technically, we reuse the probabilistic model introduced by Teney and Piater [13]. It uses the image features to define a non-parametric distribution over the image space. Although presented in a general formulation, that model has only been applied, to our knowledge, to edge segments. In this paper, we extend it to the use of dense image information, which we demonstrate using image gradients. We define appropriate kernels and sampling techniques to ensure a balance between performance and efficiency. Although the same authors proposed improvements specific to the use of edges [13] (notably weighting different parts of the model), we did not include these in the discussion in the interest of clarity. Since these improvements can significantly improve the initial detection when using edges alone, the evaluation in this paper focuses on the *improvement* brought by our novel gradient features, rather than on absolute performance values.

2 Probabilistic model of appearance

2.1 From image features to probability distributions

Our framework for object detection and recognition is based on a representation of images as continuous probability distributions of image features. This approach is used identically

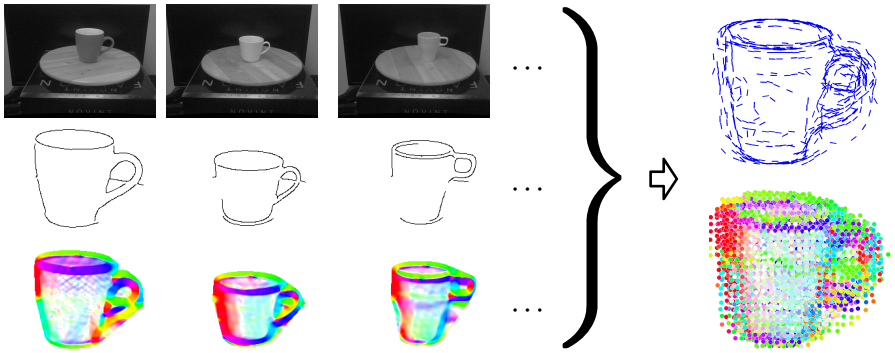


Figure 1: We model the appearance of an object with continuous probability distributions of image features, defined from one or several training examples (left). The model can be visualized by drawing samples (right) from those distributions, in this case as edge points and gradient points (hue/saturation represent respectively orientation and magnitude).

for both the example images, used for training, and for the test image of the scene in which to detect the object. Formally, considering a particular type of image features, denoted by an index k, we extract from an image such features, which form a set of observations $\mathcal{O}^k = \{x_i\}_{i=1}^{M_k}$. These image features x_i are defined on the *appearance* space $\mathcal{A}^k$, which differs between the different types of features. As detailed in Section 3, we focus on the use of two complementary types of features, namely edge points and gradient points, that we now briefly introduce to illustrate the following discussion. Edge points consist of points identified along edges in the image. Those may simply correspond to the pixels of an edge map, or to points identified with a sub-pixel accuracy. Each point is characterized by its position in the image and the local (tangent) orientation of the edge (an angle in $[0,\pi[$), so that $\mathcal{A}^{edges} = \mathbb{R}^2 \times S_1^+$. The gradient points are obtained by convolving the image with a derivative of a Gaussian; this produces a feature point for every pixel of the image, characterized by, in addition to its position, the orientation (in $[0,2\pi[$) and the magnitude of the gradient, so that $\mathcal{A}^{gradients} = \mathbb{R}^2 \times S_1 \times \mathbb{R}^+$.

The motivation for representing images as probability distributions is twofold. First, this representation accounts for the inevitable uncertainty of the description of any single image, due e.g. to image noise, quantization errors, uncertainty during feature extraction, etc. Secondly, it provides, within the same formulation, a way of modeling variability in appearance of an object or object category, e.g. given several different examples of this category. We define continuous probability distributions over the appearance space of image features and represent them in a non-parametric manner, using kernel density estimation (KDE) with the image features as supporting particles. For a particular type of image features k, we use the features $\mathcal{O}^k$ to define the distribution

$$\phi_{\mathcal{O}^k}(x) = \frac{1}{|\mathcal{O}^k|} \sum_{x_i \in \mathcal{O}^k} \mathsf{K}^k(x, x_i)\,, \tag{1}$$

with $x \in \mathcal{A}^k$, and $\mathsf{K}^k(\cdot,\cdot)$ a kernel suited to the type k of image features (see Section 3). Practically, this definition gives us a probability density function that can be easily evaluated for any x. For example, in the case of edges, we can retrieve the probability of observing a horizontal edge at a specific location in the image. Assigning kernels to each image feature

accounts for the uncertainty in the description of the features, similar to some "blurring" in the appearance space $\mathcal{A}^k$. Building a model of an object *category*, using several, slightly different training examples, is done using the exact same formulation: image features are extracted from all training examples (assuming they are aligned and at the same scale), and their union forms the set $\mathcal{O}^k$, so that the resulting distribution $\phi_{\mathcal{O}^k}$ is representative of the distribution of features among all those training examples together.

2.2 Use of probabilistic models for detection and recognition

In order to use our representation of images as distributions in the context of object recognition, we must provide a way to measure the similarity between a test image and a training template. Assuming those are respectively defined by ϕ_1^k and ϕ_2^k, we measure their similarity with

$$\int_{\mathcal{A}^k} \phi_1^k(x) \quad \phi_2^k(x) \; \mathrm{d}x \; . \tag{2}$$

In the context of object detection in an image, the template may appear under any similarity transformation (translations, rotations, scale changes) that we denote by w, and which are trivially hard-coded in a function $t_w(x)$. Accounting for such transformations, the similarity function then becomes the cross-correlation

$$(\phi_1^k \star \phi_2^k)(w) \; = \; \int_{\mathcal{A}^k} \phi_1^k(x) \quad t_w\big(\phi_2^k(x)\big) \; \mathrm{d}x \; . \tag{3}$$

Taking into account several types of image features ($k = 1..K$), we use the product over k of this expression, and the final similarity measure between two images is thus given by

$$\prod_k \; (\phi_1^k \star \phi_2^k)(w) \; . \tag{4}$$

Evaluating the similarity function for a given w The similarity measure can easily be evaluated for a specific w (a location, scale and orientation in the image), which gives a recognition score of a training template in an image. The integral of Eq. 3 is approximated with Monte Carlo integration. This involves drawing samples x_ℓ from the distribution ϕ_2^k (see Section 2.3), and simply computing the following sum:

$$(\phi_1^k \star \phi_2^k)(w) \; \approx \; \frac{1}{L} \sum_\ell^L \phi_1^k\big(t_w(x_\ell)\big) \; . \tag{5}$$

Identifying local maxima of the similarity function Solving the problem of object localization amounts to maximizing our similarity measure, i.e. finding $\arg\max_w \prod_k (\phi_1^k \star \phi_2^k)(w)$. We use for this purpose the voting algorithm proposed in [13]. It is reminiscent of the classical Hough scheme, but it also approximates our similarity measure as defined above. In practice, that algorithm is used to identify local maxima in the space of w using one type of image features at a time. Indeed, in the problem of localization in an image, the meaningful optima of the full similarity function (using several types of image features, Eq. 4) will also correspond to local optima for each type features alone. For efficiency, we therefore run this procedure using the sparser *edge* features, and then compute the exact similarity scores with edges and gradients together (Eq. 4), at those discrete points proposed by the voting algorithm. It is however interesting to note that dense features can be also used *alone* with this voting procedure, e.g. if no edges are available, as demonstrated with the concave/convex dots (see Section 4.1).

2.3 Sampling from distributions of image features

As presented above, using our representation of images as distributions involves drawing samples from those distributions. The number of samples used (L in Eq. 5) is a parameter to be chosen. The most accurate results can easily be obtained with large numbers of samples, but at the cost of large computing requirements. The basic method for sampling from our distributions, which are defined with KDE, involves selecting one of the particles at random, then drawing a sample from the kernel centered on that particle. In the case of models of object categories, or of objects for which we use a large number of different training examples, some particles account for "noise" in the distribution. This noise is due to non-meaningful variations of appearance among the training examples that we wish *not* to actually capture in the model. As those particles typically make up only a small fraction of the model, the proposed probabilistic algorithms are not detrimentally affected as long as the number of samples used is large. We however found it useful to use an alternate sampling method, which focuses on the main modes of the distribution, to provide more consistent results when using smaller numbers of samples, therefore leading to better performance at a lower computational cost.

The proposed method differs from the basic one (described above) in the selection of the supporting particle. Instead of choosing it uniformly at random, we weight the particles with their own likelihood under their distribution. Formally, given a set of image features $\mathcal{O}^k = \{x_i\}_{i=1}^{M_k}$, which define the distribution $\phi^k_{\mathcal{O}^k}$, we assign, to each particle x_i, the weight $\mathrm{wt}_i = \phi^k_{\mathcal{O}^k}(x_i)$. The probability of selecting a particle is then set to be proportional to its weight. Similar procedures for drawing samples from the main modes of a distribution have been previously proposed in the literature, e.g. in [2] under the name of 2-level importance sampling[1]. Fig. 9 provides visual comparisons of sampling methods.

Finally, let us remark that the computing and storage requirements for the image features may be large, but that a subsampling of the models, using the above procedure, may be performed off-line, as a preprocessing step. One therefore only has to store precomputed samples, which are then readily available at test time.

3 Application to edges and gradients as image features

The concepts and methods presented in this paper are generally applicable to various types of image features. This section describes in more detail the specifics of the two complementary types of image features used in our implementation: edge points and gradient points (see Table 1 for details).

Edge points We use the classical Canny detector to extract edges from an input image. Although one could then use every pixel of the resulting edge map as features, we found a good balance of performance and efficiency by following the edges and keeping points along them at regular distances (in the order of 5 pixels). The local tangent orientation of an edge is determined as it is followed, and assigned to the resulting image features. The kernel used to compare edge points (Table 1) takes into account both the position of the features (within a Gaussian kernel) and their orientation (within a von Mises kernel, which is similar to the wrapped Normal distribution).

[1] Formulated using importance sampling, the proposed technique corresponds to using ϕ as the proposal distribution to sample from a distribution ϕ' in which the probability densities would have been squared.

Edge points		
$x = (x^{\text{pos}}, x^{\text{ori}})$	with $x^{\text{pos}} \in \mathbb{R}^2$	location in image
	$x^{\text{ori}} \in S_1^+$	orientation
$\mathsf{K}(x_1, x_2) = \mathcal{N}(x_1^{\text{pos}}; x_2^{\text{pos}}, \sigma^{\text{pos}})\ \text{VM}^+(x_1^{\text{ori}}; x_2^{\text{ori}}, \kappa^{\text{ori}})$		

Gradient points (undirected, directed, semi-directed)		
$x = (x^{\text{pos}}, x^{\text{ori}}, x^{\text{mag}})$	with $x^{\text{pos}} \in \mathbb{R}^2$	location in image
	$x^{\text{ori}} \in S_1$	orientation
	$x^{\text{mag}} \in \mathbb{R}^+$	magnitude
$\mathsf{K}(x_1, x_2) = \frac{1}{C}\ \mathcal{N}(x_1^{\text{pos}}; x_2^{\text{pos}}, \sigma^{\text{pos}})$		with C a normalization constant
$\max\Big(\mathcal{N}(x_1^{\text{mag}}; 0, \sigma^{\text{mag}})\ \mathcal{N}(x_2^{\text{mag}}; 0, \sigma^{\text{mag}}),$		
$\mathcal{N}(x_1^{\text{mag}}; 255, \sigma^{\text{mag}})\ \mathcal{N}(x_2^{\text{mag}}; 255, \sigma^{\text{mag}})\ \text{VM}^+(x_1^{\text{ori}}; x_2^{\text{ori}}, \kappa^{\text{ori}})\ \big(1 - \frac{1}{2}\lvert\sin(x_1^{\text{ori}}) - \sin(x_2^{\text{ori}})\rvert\big)\Big)$		

Table 1: Formal definition of kernels for edge and gradient features. The notations VM and VM^+ denote von Mises kernels respectively on $S_1 = [0, 2\pi[$ and $S_1^+ = [0, \pi[$.

Gradient points Our gradient features are designed to capture image information corresponding to homogeneous regions in the image, and regions of slowly varying gradients, due e.g. to shading on a smooth surface. It is easy to see how this information is complementary to the edges, which rather capture sharp transitions. Edges, of course, are also identified using gradients, and it is thus desirable to ensure that the same information is not captured redundantly by both types of features. We extract gradients by first convolving the image with derivative-of-Gaussian filters in horizontal and vertical orientations. The result is then used to determine the magnitude and orientation (an angle in $[0, 2\pi[$) of the gradient for every pixel of the image. To exclude the information already captured by edge features, we simply discard the gradients in the regions close to edges. Finally, and as motivated above, the resulting features are selected as pixels at regular locations in the image.

Our kernel for gradient features is defined in Table 1. It favors matches between either (1) gradients of both low magnitudes, without taking orientation into account (for homogeneous regions in the image), or (2) gradients of both significant magnitudes and similar orientations. In the latter case, we evaluated the use of the orientation in three different ways: undirected, directed or semi-directed. In the *undirected* manner, the orientation of the gradients is compared only on the half-circle. Two horizontal gradients, from black to white and from black to white would thus be considered identical. In the *directed* manner, their orientation in that case would be considered opposite, offering a more discriminant comparison. However, as a downside to this second option, the direction of gradients caused by shading on smooth surfaces may not always be relevant. We hypothesized (and experimentally verified, see Section 4.1) that gradient direction is most relevant as the gradient is vertical, as lighting coming from above the scene is a common prior that we can use, whereas the difference in direction between gradients appearing horizontally in the image are not as meaningful. This is captured by the *semi-directed* formulation presented in Table 1.

4 Experimental evaluation

We evaluate our contributions on both new and existing datasets. The key advantage of using our dense gradient features is to solve ambiguous cases where edges alone do not provide sufficient information to determine the presence (in clutter) of an object or its pose (due to

equivocal appearance under for different viewpoints). Such peculiar cases are arguably infrequent and were often simply avoided in existing datasets for pose estimation. We therefore resorted to synthetic test data, which allows us to demonstrate some telling examples and to simulate varying illumination to evaluate the behavior of the proposed gradient features.

We also provide results for detection and recognition on existing datasets. Note that some improvements specific to edges, applicable within the proposed framework (e.g. weighting the edge points), were shown to significantly improve the performance of initial detections with edges alone [13], but were not included in this paper for the sake of clarity. Therefore, our evaluation focuses on the *improvement* achieved by the novel gradient features, rather than on absolute performance values. In general, the method did not prove too sensitive to choices of parameters. The bandwidth of the kernels (Table 1) are set to allow variability in orientation of about 30° (κ^{ori}) and in position of about 5 pixels (σ^{pos}). The size of the filter used to extract gradients is set as a fraction of the object size, in the order of $\sigma = 2-3$ pixels.

Note that all illustrations that include gradients use the following color coding: the orientation of the gradient is mapped from $[0, 2\pi[$ to the hue channel and its magnitude is used to set the saturation channel proportionally.

4.1 Demonstration examples with synthetic data

The following experiments were designed to provide insight into the behaviour of the proposed gradient features, in particular under different conditions of illumination. Synthetic images were generated with raytracing software [8] using global illumination maps (Fig. 2) captured in real environments [1, 14], which allow the simulation of complex realistic lighting conditions. All images in these experiments assume a camera looking straight ahead, i.e. with the optical axis horizontal. Experiments both consist in 2-way classification tasks.

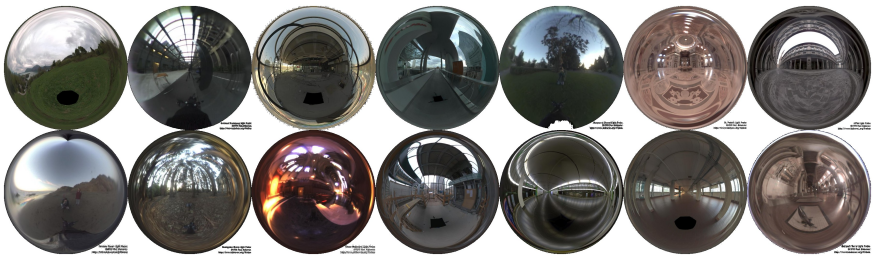


Figure 2: "Light probe" illumination maps, used to generate the synthetic scenes. These high dynamic range images specify the incident illumination coming from all directions onto the scene.

Concave/convex dots This experiment replicates a well-known visual illusion, where one is looking perpendicularly at surface that presents either a bulge (convex) or a dip (concave). No other clue than the shading on the surface can help the observer to determine whether this "dot" is convex or concave, and, if the conditions of illumination are not known, the ambiguity cannot be solved. A human observer will usually assume an illumination coming from above, as is the most common in the real world, and infer the geometry of the dot accordingly. We perform experiments with a leave-one-out cross-validation. We trained two models with our recognition system, with images of respectively concave and convex

dots, under 13 different conditions of illumination. The testing then involved determining the shape of the dot in an image with a 14^{th} unseen illumination. The highest similarity score, with either the concave or the convex model, gives the result of the classification. The experiments showed that the shape of the surface could be determined using our gradient features as long as the illuminations were realistic, and that our model could thus properly capture this "light from above" prior. Detailed results are reported in Figures 3–5.

Hollow can We designed another test case where the use of gradient features could be tested. The object is a cylinder with one hollow and one closed end. This "can" appears under different elevations/viewpoints (one model is trained for each viewpoint), and may be stand either upright or upside down, which both look identical if one considers only image edges. Gradients, on the contrary, appear mostly homogeneous on the flat surface of the closed end, and present varying orientations on the other, hollow end. Training and test images were again generated using the 14 illumination patterns of Fig. 2. We investigated the effectiveness of the three proposed matching criterion of gradients (Section 3): we found the *semi-directed* formulation to be most helpful with fewer training examples, when the actual variability in horizontal (but not vertical) gradients was not well represented within the limited training examples. Further details and results are reported in Figures 6 and 7.

4.2 Tabletop dataset

We evaluate the performance of our method for object detection in clutter with real images using the "tabletop" dataset of [12]. It features a total of 30 objects from 3 categories: computer mice, mugs and staplers. Training uses the part of the dataset with objects appearing on a turntable under known viewpoints ("Table-Top-Pose"; see Fig. 1). A model is learned for each viewpoint of each object category. Testing is performed on scenes ("Table-Top-Local") containing one or several instances of the objects in a cluttered office environment. We perform detection of each object category separately (detecting any viewpoint-model of that category), and measure performance with the well-known Pascal VOC criterion (bounding boxes overlap). We report results in Fig. 8 as precision/recall curves (after removing overlapping detections). The use of gradients brings a significant improvement over edges only, in particular on cups, the shape of which produces very characteristic shading patterns. The improvement is also present with the mice and staplers, but not so dramatic. The staplers produce less characteristic gradients due to the small flat surfaces, already well defined by edges alone. The mice are very diverse in shape, but observed under fixed lighting conditions in the training images, which moreover produces specular highlights, which do not appear in the testing images. We believe more significant improvement would be brought by using gradients if the training images presented varied lighting conditions.

4.3 3D Object dataset

We finally evaluate our method on the "cars" category of the 3D object dataset of Savarese *et al*. We use similar conditions and evaluation criteria as [10], a 5/5 training/test split with the 2 largest scales of the dataset, and learn a model for each viewpoint (8 angles, 3 heights). We perform detection for each of those models, which give detection candidates for each specific viewpoint (pose). We report our results in the way of [10] (Fig. 10) and show comparable performance, with, again, a significant improvement by using gradients over edges alone.

	Success rate	Score separability (lower ≈ better): ratio $similarityScore_{incorrect}$ / $similarityScore_{correct}$													
Illuminations maps															
Images (concave)															
Images (convex)	Avg.	Avg.													
Undir. grad.	54%	1.00	.9	1.1	.9	.9	1.0	1.0	1.1	1.0	1.0	1.1	.9	.9	.9	1.1
Dir. grad.	**100%**	**0.47**	.4	.5	.5	.4	.5	.4	.5	.5	.4	.5	.4	.6	.7	.4
Semi-dir. grad.	**100%**	0.48	.4	.5	.5	.5	.5	.4	.5	.5	.4	.5	.4	.6	.7	.5

Figure 3: Concave/convex dots dataset. Results of recognition from a leave-one-out evaluation. The separability of correct/incorrect scores is interestingly worse with uncommon illumination patterns (red), with the light coming mostly from the sides of the scene.

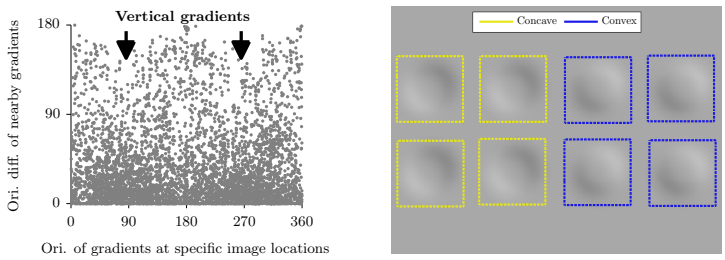


Figure 4: Concave/convex dots dataset. Additional results on recognition experiments. Left: many samples (gradient points) are drawn from the model, and we plot the orientation of those samples *versus* the difference in orientation of nearby samples. Variability in orientation is observed to be *least* when gradients are vertical, which justifies our *semi-directed* matching of gradients. Right: demonstration test scene, where we perform detection using the models for concave, then convex dots: all the instances are correctly detected. Note that the voting algorithm is used here with gradients directly, since no edges are present in the image.

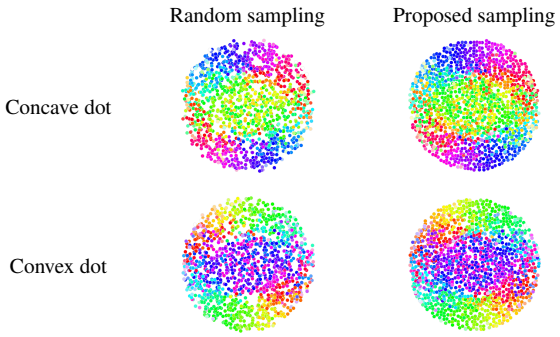


Figure 5: Concave/convex dots dataset. Learned model (of gradients) of the concave and convex dots, visualized as samples drawn from the model, using two different sampling techniques, with the same number of samples for both. Visual differences are minimal, but the proposed technique generally provides slightly better performance with fewer samples.

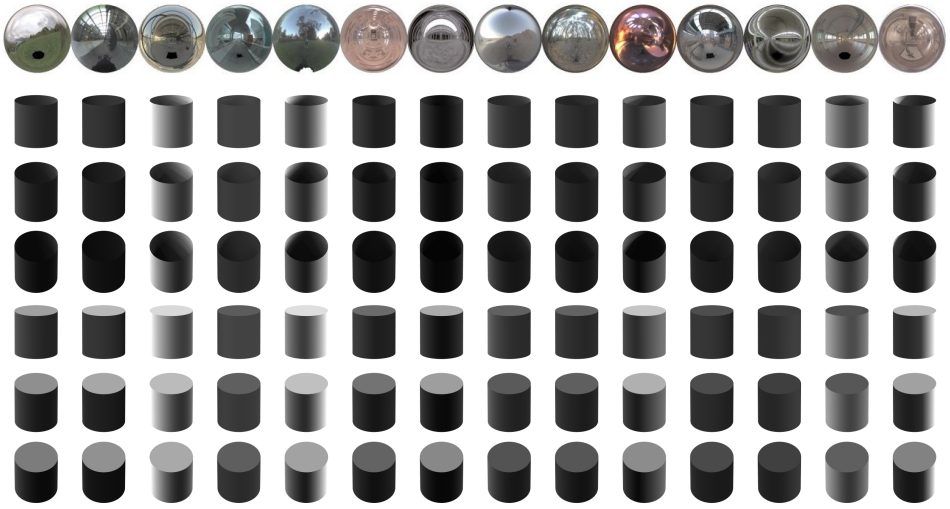


Figure 6: Hollow can dataset. All the images used for training and testing. Six different models are learned, for each elevation in upright/upside-down configurations. Testing, as reported in the paper, uses the 6 images of each illumination successively (i.e. a total of $6.14 = 84$ tests); for each test illumination, N (as reported in the evaluation) other training illuminations are selected at random to build the 6 models.

	Recognition (upright / upside-down) rate				
N. of training examples (illuminations)	2	4	6	8	13
Edges only	45%	57%	44%	49%	51%
Edges + gradients (undirected)	86%	86%	89%	82%	76%
Edges + gradients (directed)	87%	90%	91%	85%	80%
Edges + gradients (semi-directed)	**92%**	**92%**	**92%**	**85%**	**81%**

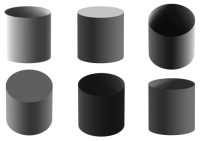


Figure 7: Hollow can dataset. Recognition results with different numbers of training illuminations (selected at random from the 14 of Fig. 2). The *semi-directed* formulation is most helpful with fewer training examples, when the actual variability in horizontal (but not vertical) gradients is not well represented within the limited training examples. Right: sample images of the can in upright and upside-down (top-middle, bottom-left) configurations.

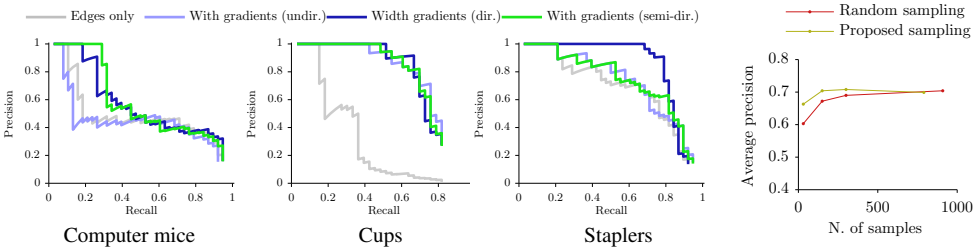


Figure 8: Tabletop dataset. Left: results of object detection on the tabletop dataset. Using gradients brings a large improvement with the cups, thanks to characteristic shading patterns due to their shape. Right: influence of the number of samples for the cups. With small numbers of samples, the proposed sampling method provides more consistent results than the basic method, therefore leading to slightly better performance at a lower computational cost.

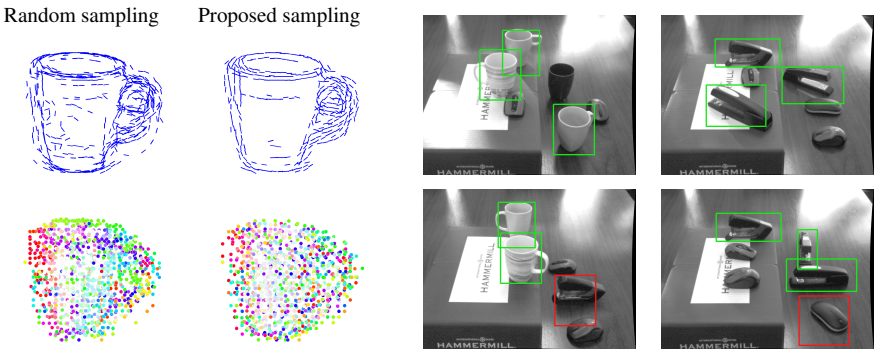


Figure 9: Tabletop dataset. Left: learned model of the cup (for a specific viewpoint) of the tabletop dataset, visualized as samples drawn from the model, using two different sampling techniques, with the same number of samples for both. The proposed technique generally provides slightly better performance with fewer samples. Right: visualization of detections of cups and staplers.

	Detection (AP)	Pose (MPPE)
Savarese *et al.* [10]	70.0%	–
Baseline comparison: edges only	63.8%	88.1%
With gradients (undirected)	66.9%	91.9%
With gradients (directed)	71.9%	89.6%
With gradients (semi-directed)	71.3%	90.4%

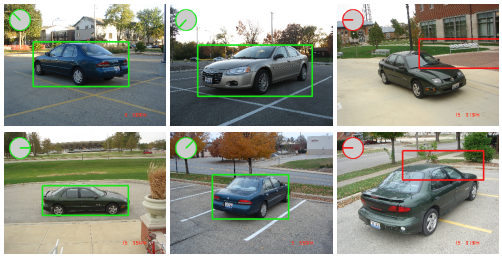


Figure 10: 3D Object dataset. Results of detection and discrete pose recognition on cars.

5 Conclusions

We extended an existing probabilistic model of appearance to dense image features. We used, within that framework, image gradients, in addition to classical edges, and observed a significant improvement on object detection and recognition. We verified the relevance of gradient orientation *and* direction, but found the best results with a formulation that takes direction into account only for mostly-vertical gradients. This encodes our natural prior for light coming from above the scene, and invariance to left/right differences. Future developments could integrate refinements such as the weighting of model features [13] to further improve recognition performance, together with efficiency-oriented algorithmic developments.

References

[1] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH*, pages 189–198. ACM, 1998.

[2] Renaud Detry, Nicolas Pugeault, and Justus Piater. A probabilistic framework for 3D

visual object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1790–1803, 2009.

[3] Staffan Ekvall, Frank Hoffmann, and Danica Kragic. Object recognition and pose estimation for robotic manipulation using color cooccurrence histograms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[4] Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, and Rama Chellappa. Fast directional chamfer matching. In *Computer Vision and Pattern Recognition*, pages 1696–1703, 2010.

[5] Pradit Mittrapiyanuruk, Guilherme N. DeSouza, and Avinash C. Kak. Calculating the 3D-pose of rigid-objects using active appearance models. In *IEEE International Conference on Robotics and Automation*, pages 5147–5152, 2004.

[6] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision*, 2008.

[7] Arthur R. Pope and David G. Lowe. Probabilistic models of appearance for 3D object recognition. *International Journal of Computer Vision*, 2000.

[8] Rhinoceros. Robert McNeel and Associates. URL http://www.rhino3d.com.

[9] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3):231–259, 2006.

[10] S. Savarese and Li Fei-Fei. 3D generic object categorization, localization and pose estimation. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[11] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1270–1281, 2008.

[12] Min Sun, Gary Bradski, Bing-Xin Xu, and Silvio Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *European conference on Computer vision*, pages 658–671. Springer-Verlag, 2010.

[13] Damien Teney and Justus Piater. Continuous pose estimation in 2D images at instance and category levels. In *Computer and Robot Vision*, 2013.

[14] Bernhard Vogl. Light probes, 2010. URL http://dativ.at/lightprobes/.

Chapter 8

Unified presentation and evaluation of our contributions on exemplar-based recognition

This chapter presents a unified formulation of our most interesting contributions on exemplar-based object recognition and pose estimation, introduced in Chapters 4–7. These contributions, although they were originally presented in separate conference papers, are now integrated, all together, in a common coherent framework. More explicitly, with respect to these previous chapters, we now combine the following points.

- The representation of test and training images as probability distributions of image features (Chapter 4).
- The modeling of appearance of object categories through the use of distributions of features (Chapter 5).
- The formulation of the similarity between a test image and an object model, and of the localization problem as a cross-correlation between distributions of features (Chapter 4).
- The voting-based algorithm to approximate the similarity measure, and thereby to efficiently perform detection in cluttered images (Chapter 5).
- The principle of assigning different weights to the image features of the training data (as in Chapter 5, although in a slightly different manner, see below).
- The explicit modeling of changes of appearances as deformations between discrete training views, that we then use for continuous pose estimation (Chapter 6).
- The use of both edges (Chapter 5) and coarse-scale gradients (Chapter 7) as image features.

In addition to these components already introduced in preceding chapters, we present the following new contributions.

- The application of the modeling of deformations (Chapter 6) to object categories.

- A resampling procedure particularly suitable to category models learned from noisy or cluttered examples, which focuses on the main modes of the distribution.
- An iterative procedure for learning weights on training data from negative examples, that leads to improved detection rates in cluttered images.
- A software implementation of the overall method that exploits parallelization on Graphical Processing Units (GPUs); although it is not presented as a central contribution in the following paper, it is of considerable practical interest as it speeds up most computations by a factor in the order of 20, and widens the range of applicability of the method.
- An extensive evaluation of the proposed framework on a number of tasks and benchmark datasets. It demonstrates performance on classical tasks well above baseline methods, often competitive with more complex methods individually designed for some of these specific tasks. In comparison, the proposed framework is more widely applicable. We also demonstrate the unique capabilities offered by our framework with particular types of objects, of primitive shapes or with little texture, and thus hard or impossible to handle with traditional edge-based methods.

At the time of the writing of this thesis, the journal paper included in the next pages has been submitted to *Computer Vision and Image Understanding* and is still under review.

Multi-view Feature Distributions for Object Detection and Continuous Pose Estimation

Damien Teney*

Montefiore Institute
University of Liège
Grande Traverse 10, B-4000 Liège, Belgium

Justus Piater

Intelligent and Interactive Systems
University of Innsbruck
Technikerstrasse 21a, A-6020 Innsbruck, Austria

Abstract

This paper presents a multi-view model of object categories, generally applicable to virtually any type of image features, and methods to efficiently perform, in a unified manner, detection, localization and continuous pose estimation in novel scenes. We represent appearance as distributions of low-level, fine-grained image features. Multiview models encode the appearance of objects at discrete viewpoints, and, in addition, how these viewpoints deform into one another as the viewpoint continuously varies. Using a measure of similarity between an arbitrary test image and such a model at chosen viewpoints, we perform the tasks mentionned above with a same, common method. We leverage the simplicity of low-level image features, such as points extracted along edges, or coarse-scale gradients extracted densely over the images, which we use to build probabilistic templates, i.e. distributions of features, learned from one or several training examples. We efficiently handle these distributions with probabilistic techniques such as kernel density destimation, Monte Carlo integration and importance sampling. In addition to a rigorous but straighforward formulation of the proposed framework, we provide extensive evaluation on a number of very different benchmark datasets. We demonstrate performance on the "ETHZ Shape" dataset, with single (hand-drawn) and multiple training examples, well above baseline methods, on par with a number of more task-specific methods. We obtain remarkable performance on the recognition of more complex objects, notably the cars of the "3D Object" dataset of Savarese *et al.* with detection rates of 92.5% and an accuracy in pose estimation of 91%. We perform better than the state-of-the-art on continuous pose estimation with the "rotating cars" dataset of Ozuysal *et al.* . We also demonstrate particular capabilities with a novel dataset featuring non-textured objects of undistinctive shapes, the pose of which can only be determined from shading, captured here by coarse scale intensity gradients.

*Corresponding author.
Email addresses: damien.teney@ulg.ac.be (Damien Teney), justus.piater@uibk.ac.be (Justus Piater)

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.87)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

1. Introduction and related work

This paper is primarily concerned with the recognition of object categories in 2D images. At the same time, we are also interested in identifying the pose, or 3D orientation of the objects, and we argue that those tasks are two sides of a same problem. Indeed, for example in the context of interactive applications, the two tasks are not clearly separated, and we thus tackle them with a unified approach. In addition, one cannot, in general, recognize and identify the pose of objects from just one type of image information, e.g. silhouette and edges, to cite a common example. Additional visual clues may be necessary, such as the shading onto the surfaces of the objects. A generally-applicable recognition method must thus be versatile in this regard. A key point of our contributions is to provide techniques generally applicable, even to low-level, dense and/or non-descriptive image features. Together with the general multi-view model of appearance, we provide straightforward methods to measure its similarity with a novel test image, enabling us to perform detection, recognition and pose estimation in a unified manner. The following paragraphs present the principal motivations and key points of the method, comparing them to existing related work. Parts of the contributions of this article were introduced in earlier publications [1, 2].

1.1. Tasks considered

In the context of recognition of objects in 2D images, the following tasks are usually considered, and often seen as separate research problems. They are however closely related, and we handle them all with the same model and methods. Notably, we do not train discriminative models, which is the usual approach for the classification tasks.

Localization The goal is to identify the parts of the test image that belong to the object of interest, versus the parts of the image that correspond to background clutter. The result of localization is typically a set of bounding boxes, which encircle candidate objects in the image, each accompanied with a detection score. We handle this task with an algorithm similar to the generalized Hough voting scheme. The model of the object can be learned from *one* or *several* training examples: we handle both cases identically by modeling distributions of features through kernel density estimation (see Section 2).

Detection One must decide whether the object of interest appears in the test image or not. One usually does this together with localization (both terms being then used interchangeably), by setting a threshold on scores of localizations to obtain a binary result for detections.

Classification (among objects or among discrete poses) One must determine which object or which viewpoint among learned ones appears in the image. This traditionally involves learning discriminative classifiers. In our method however, we simply build generative models independently for each learned object or viewpoint, and determine the best match from the similarity measured between the test image and one of those models.

Continuous viewpoint (pose) estimation This more challenging task is handled by extending our generative models to also synthesize unseen (untrained) viewpoints.

1.2. Modeling object appearance

The method for performing recognition of objects in 2D images depends heavily on the internal representation chosen to model the appearance of those objects. We are interested in building models of appearance for object *categories* (or "classes") rather than specific instances, thus capable

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.88)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

of recognizing, to some extent, unseen objects that are similar to a category learned from a few training examples. The goal is for example to train the system with a set of different cars, then to recognize the pose of a new, unseen car. The categories in such a scenario are defined implicitly by the training instances used as examples. In the proposed approach, the appearance of the object under a specific viewpoint is modeled as the distribution of low-level image features, represented, in a non-parametric manner, by the actual image features of one or several training images of the objects under that specific viewpoint. We therefore handle variability in appearance in a probabilistic way; this variability among the training examples can equally come from different objects of a same category, or from variations of appearance of a unique object, e.g. observed under different conditions of illuminations.

1.3. 2D and 3D object models

While some methods seek to build an explicit, geometrical, 3D model of the objects [3], we rather choose to use the 2D appearance alone, and train our model with simple example images. The motive for this choice is to handle more easily and naturally the variability within categories, both in appearance *and shape*. Some methods relying on rough 3D models have included possible variations in appearance [4, 5], but this variability is limited in regards with shape. One exception is the work of Glasner *et al.* [6], which uses structure-from-motion to reconstruct accurate 3D models from the training images. They then account for within-category variability simply by merging multiple exemplars in their non-parametric model, in a fashion very similar to the one we use (for our 2D training examples). One drawback of the approach is, however, the initial need for a large number views, in order to reconstruct the 3D models. In comparison, our exemplar-based model can use an arbitrary number of views, which do not need to overlap, and the model can be incrementally updated as more views become available.

1.4. Object localization and detection

Object localization and detection among clutter is commonly achieved with variants of either the "sliding window" or the "Hough voting" approaches. The former (used e.g. in [7]) uses a binary classifier, which is evaluated on a uniform sample of image locations and scales. Such an exhaustive search may prove computationally expensive, and many heuristics have been proposed to alleviate this issue [8]: salient regions, coarse-to-fine-search, etc. Voting techniques based on the well-known generalized Hough transform [9] provides another way to alleviate the complexity issue. Probabilistic formulations of this voting technique have been proposed through the implicit shape models [10, 11]. Our algorithm for detection uses this voting scheme, applied to low-level, dense image features. Hough voting was exended to discriminative framework by Maji and Malik [12], by computing optimal weights to the image features of the model. They obtained excellent results, further improved by a subsequent verification step, in which the initial detections are rescored by an SVM-based classifier. We reuse this idea of weighting parts of the learned model; the exact procedure is slightly different, and suited to our non-discriminative features. Although not a central element of our contributions, we will show that this weighting often brings substantial improvements.

1.5. Choice of image features

The type of image features used to encode the appearance of the objects is a crucial choice. Some methods historically used of the appearance of the object as a whole [13–15], but with the common downsides of poor robustness to occlusions and a need for large numbers of training views. At the opposite end, *feature-based* methods have relied on "interest points", precisely located in the images, and characterized by hand-designed descriptors of local appearance, such as SIFT descriptors [16].

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.89)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

Those discrete points can then be matched between the test image and the training examples [3]. While this approach has proved to be highly successful and efficient in many cases, the extraction of such discriminant image features cannot be relied upon in general cases, as it often fails with non-textured objects. The basic approach also does not readily extend to variability within categories. A recent trend is to describe image contents with similar descriptors of appearance over a *dense* grid across the image, such as done by the successful histograms of oriented gradients (HOG) [7], also used within the state-of-the-art detector of Felzenszwalb *et al.* [17]. The idea behind those descriptors is to capture statistics or distributions of primitive characteristics (such as intensity gradients) over local image regions. We believe that this approach is indeed the most generally-applicable one, and is the central motivation for our technique. Similarly to, e.g. HOGs, our "distributions of features" capture local statistics densely over the images, but we do not depend on hand-designed descriptors, and we offer a unique formulation suitable to different types of image features. Another notable difference of our method with HOGs is to use gradients extracted at a coarse scale, intended to capture shape (rather than pure appearance) of smooth surfaces, whereas HOGs were most successful with gradients extracted at a much smaller scale, thus essentially capturing sharp transitions like edges.

Most current, state-of-the-art methods for object recognition rely on the use of image edges (e.g. [18, 19], among many others), seen as an efficient representation of the silhouette and shape of objects. The typical technique basically consists in building intermediate representations such as contour fragments, which can then be matched discriminatively between training and test images, and used e.g. in a Hough voting scheme. Our approach, which leverages the simplicity of low-level, fine-grained image features, can be applied to edges by considering all edge pixels of the image as features. At the cost of higher computational costs, this approach leads to excellent results as well, while satisfying our aim for a general and straightforward formulation.

A large area of research has focused on the modeling and detection of deformable shapes (see [18] for a review). Interestingly, our simple approach proves competitive with some of those techniques, as demonstrated on the ETHZ shape dataset. Although we neither model continuous contours nor their variations explicitly, our low level features (edge points) can encode similar variations to some degree. Another advantage of our our method is its ability to learn shape models similarly from a single or multiple examples, and from only *loosely* segmented images (with a bounding box). Such capabilities are not commonplace in the domain of shape matching, but were also offered by the work of Ferrari *et al.* [18].

Finally, our capability of handling *dense* image features is demonstrated and used with great advantage with intensity gradients, extracted at a coarse-scale over the whole images. Using such gradients provides unique capabilities, as it allows one (1) to effectively handle non-textured objects (see Section 5.6), and, even more importantly, (2) to resolve cases where edges alone would only offer ambiguous information on the presence or the pose of an object in a scene. Indeed, the shading over homogeneous surfaces, captured by such gradients, may sometimes be the sole relevant clue, in particular to identify the exact pose of certain objects, or, for example, to differentiate between hollow versus full objects of similar shapes (see our experiments in Section 5.6).

1.6. Multiview models of appearance and pose estimation

Object recognition with 2D training examples typically uses viewpoint-specific models, e.g. a model for cars seen from the front, and another for cars seen from the side. Recent contributions have included more and more techniques that handle multiple registered training viewpoints. The object in the test image is then matched against one of these viewpoints and allows performing a coarse estimation of its pose (or 3D orientation) also called *pose classification*. We refer to this basic approach as a "nearest-neighbour" pose estimation. Some applications (robotic interaction

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.90)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

and grasping for example) require however a more precise estimation of the pose [20, 21]. This capability was commonly reserved to recognition methods using 3D object models. As discussed above though, they do not cope well with object *categories*, which are clearly very challenging with regards to the task of pose estimation. Few appearance-based methods have been designed to provide this capability [14]. Most recent multiview models of appearance consider the different training viewpoints independently [21–25], while others try to match and link features across viewpoints [26–28]. Savarese *et al.* [27], for example, model an object as a collection of planar parts that can appear in different views. We follow an intermediate approach, by storing independently the image features that make up the different views, but we also store, along with every each image feature, how its appearance varies with respect to the pose of the object. The multiview models mentioned above only performed localization and classification such as "frontal view" or "side view", whereas we allow precise, *continuous* pose estimation.

Simple techniques have been proposed to improve the precision of nearest-neighbour pose classification. They typically involve voting in the 3D pose space followed by averaging [21] or probabilistic smoothing schemes [1, 25, 29], leading to a precision beyond the resolution of viewpoints given as training examples. While those simple techniques have sometimes given very interesting results, we rather chose, in the work presented here, to explicitly detect, and include in the model, the changes of appearance between the discrete viewpoints seen during training (practically, how image features translate in the image, and thus how the appearance "deforms" between neighbouring viewpoints). This information extends our generative model, which can now synthesize arbitrary, untrained viewpoints. We can then finely optimize the 3D pose, starting from the initial nearest-neighbour estimates. Let us mention the work of Torki and Elgammal [30]. In their radically different approach to appearance-based pose estimation, they learn a direct regression from local image features to the pose of the object. This original approach recovers a precise pose, but cannot handle significant clutter or occlusions, and the accurate pose estimation depends on the (supervised) enforcement of a one-dimensional manifold constraint (corresponding to the 1D rotation of the object in the training examples). It is not clear how that approach would extend to the estimation of the full 3D pose of an object.

1.7. Synthesis of novel viewpoints

During an off-line training phase, we use an optical flow algorithm between pairs of images to detect how the appearance of each training object varies between these viewpoints. The image features extracted from one of these images can then ben "deformed" into the other, and the interpolation for an intermediate viewpoint is also straightforward. We thus obtain a generative model of appearance that can synthesize the appearance of the object in any (possibly unseen) viewpoint. This procedure is related to the technique of *morphing* in computer graphics [31–33], with the difference that we are considering arbitrary numbers of input views, and we do *not* rely on established correspondences between specific landmarks of the input views. This similarly contrasts with the competing method of Savarese *et al.* [34], which does use specific correspondences between nearby views. Our advantage is to handle non-textured objects with little detail. Although some global consistency in the detected deformations is enforced by the optical flow algorithm, each image feature independently stores its possible deformations. This does not limit the model to a particular class of transformations. In comparison, Savarese *et al.* [34] specifically models *affine* transformations of object parts, assuming that objects are made of large planar parts. We also use a *sparse* set of training views (typically spaced about 20° apart on the viewing sphere) and do not require videos or dense sequences of images to track features between frames, as opposed to Sun *et al.* [35].

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.91)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

1.8. Summary of contributions

Our main contributions can be summarized in the following points.

1. We present a general framework for modeling the appearance of objects and object categories, suitable to virtually any type of image features, applicable for detection and recognition without relying on hand-designed local visual descriptors, while still providing performance and efficiency on par with state-of-the-art — arguably more complex — methods.

2. We show how to handle dense, unmatchable image features, such as coarse-scale intensity gradients. This ultimately enables the method to recognize objects without texture, and to handle cases where shading constitutes the sole source of unambiguous visual information.

3. We provide a technique for identifying, and storing, within a multiview model of appearance, how the appearance varies between discrete training viewpoints. This ultimately allows performing *continuous* pose estimation of an object in a novel scene, without relying on an explicit 3D model of the object. This also readily applies to object *categories*, and not only to specific objects.

2. Probabilistic model of appearance

This section presents our model of appearance with a bottom-up description. We start by turning a set of image features of a given image into a "distribution of features", then use those representations to form our model that includes several viewpoints, and possibly several training examples for each viewpoint. We finally show how to detect and recognize those training views in a novel test image.

2.1. From image features to probability distributions

Our approach is based on a representation of images as continuous probability distributions of image features. The motivation for representing images as distributions is twofold. First, this representation accounts for the inevitable uncertainty of the description of any single image, due e.g. to image noise, quantization errors, uncertainty during feature extraction, etc. Secondly, it also provides, as we will see in the next section, a way of modeling variability in appearance of an object or object category, e.g. given several different examples of this category. It will also give us a more abstract representation of the images that is convenient to manipulate with existing probabilistic techniques, and that generally applies to any type of image features. The approach is first applied and presented for a test image — in which we want to recognize the object of interest — while the next section will then apply it to the training examples.

We start off by extracting, from our test image, different types of features (detailed in Section 4.1), each type denoted by an index $f = 1 \dots F$. These can be as simple as the pixels belonging to edges (which we call "edge points"), or to the value of the intensity gradients for all pixels of the image ("gradient points"). In general, each feature x is thus characterized by (1) its position in the image, noted $x.pos$ ($\in \mathbb{R}^2$) and (2) some appearance attributes, noted $x.app$. In the case of edge points, we use, as an attribute, the local orientation of the edge (an angle in $S_1^+ = [0, \pi[$); in the case of gradient points, we use the orientation and the magnitude of the gradient. The contents of our test image form thus, for each type f of features, a set $\mathsf{test}^f = \{x_i\}_i$, with $x_i \in \mathcal{A}^f$, the domain of these features. For example with edge points, $\mathcal{A}^{\text{edges}} = \mathbb{R}^2 \times S_1^+$ (see Section 4.1 for details).

We now show how to turn such a set of discrete image features into a continuous probability distribution. We define and represent such distributions over the appearance space of image features

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.92)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

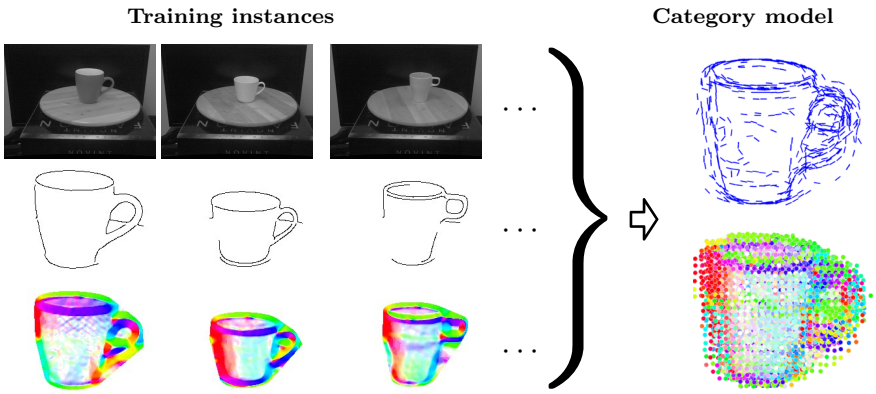


Figure 1: We model the appearance of an object with continuous probability distributions of image features, defined from one or several training examples (left). The model can be visualized by drawing samples (right) from those distributions, in this case as edge points and gradient points (hue/saturation represent respectively orientation and magnitude).

($\mathcal{A}^f$) in a non-parametric manner, through kernel density estimation (KDE). With this procedure, all image features are used as particles supporting simple kernels, the sum of which represents a complex continuous distribution. Formally, for each type of image features f, we use the set of features test^f extracted from our test image to define the distribution

$$\phi^f_{\mathsf{test}^f}(x) = \sum_{x_i \in \mathsf{test}^f} \mathrm{wt}(x_i)\ \mathcal{N}(x_i.pos; x.pos, \sigma_{pos})\ \mathsf{K}^f(x_i.app; x.app)\,, \tag{1}$$

with $x \in \mathcal{A}^f$, $\mathcal{N}$ a Gaussian kernel for the position of the features, K^f a kernel for their appearance attributes (see Section 4.1), and $\mathrm{wt}(x_i)$ the weight of the feature x_i. Those weights are set uniformly for the features of a test image, i.e. $\mathrm{wt}(x_i) = \frac{1}{|\mathsf{test}^f|}\ \forall x_i \in \mathsf{test}^f$. This representation with KDE will be reused for the *training* images, where the weights will then take a more complex form (Section 2.4). Practically, Eq. 1 gives us a probability density function that can be easily evaluated for any x. For example, in the case of edge points, we can evaluate the probability of observing a horizontal edge at a specific location in the image.

2.2. Application to object categories and to multiple views

We have represented our test image as continuous distributions of image features. We will now similarly apply that approach to the training images. Two differences are worth mentioning though.

First, we may observe the object of interest under *multiple* viewpoints. Each training image t corresponds to a viewpoint $v_t \in S^2$ (a point on the viewing sphere), and gives, a set of features $\mathsf{train}^f_{v_t}$ for each type of feature f (defined similarly to the sets test^f above). Those multiple viewpoints are considered independently at this point, and they each define distributions $\phi^f_{\mathsf{train}^f_{v_t}}$ as in Eq. 1. Only in Section 3 will we consider multiple viewpoints together, in order to perform continuous pose estimation. As a first step though, we are only interested in recognizing (approximately at least) one of the discrete viewpoints provided as the training examples.

Second, we may be provided with training images of several, different objects (object "instances") representative of an object *category*. We assume that all training images are aligned and at the

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.93)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

same scale, which can be practically done automatically as explained in Section 5. We now want our distributions of features to reflect statistics relevant to all the different training examples. This is straightforward within our formulation with a KDE: for each viewpoint v_t, we simply include, in the set of features train_v^f, the features extracted from *all* training images corresponding to that viewpoint (Fig. 1). The resulting distributions $\phi^f_{\mathsf{train}_v^f}$, as defined earlier, are then representative of the occurrence of image features among all those training examples together, and they constitute our model of appearance of an object *category*. Consequently, the appearance of that category is thus defined implicitly by the instances provided as training examples.

2.3. Use of proposed model for detection and recognition in a new image

We now would like to detect, or recognize the learned object in the test image. The solution to this task consists in the optimal set of in-plane transformations w^* (a translation, rotation and scaling in the image) and viewpoint (out-of-plane transformations) v^* ($\in S^2$), which corresponds to the training viewpoint recognized in the test image. Let us mention, as a side note, that this result (v^*, w^*) presents 6 degrees of freedom (DoF), and that it can be equally described in the image space (as we do) or in the "world" space (as Euclidean coordinates for position and orientation). The latter is usually preferred in the field of robotics, and commonly called the 6-DoF pose of the object. Both representations are however equivalent and interchangeable, provided the calibration of the camera.

We will first present how to measure the visual similarity between the test image and the learned object at a specific viewpoint and in-plane transformations. We will then provide an algorithm to identify the optimal set of such transformations, determining the local maxima of that similarity. At this point, we still consider the training viewpoints independently, and thus perform a "nearest-neighbour" classification of the viewpoint. This will serve as a starting pointer later, for a local optimization procedure to perform *continuous* pose estimation (Section 3).

Let us consider a test image is represented by the distributions of features ϕ^f_{test}, and a specific training view t represented by $\phi^f_{\mathsf{train}_v^f}$. This training view may appear in the test image under any similarity transformations w (in-plane translation, rotation, scaling), trivially applied by a function $\mathrm{transform}_w(x)$. Accounting for such transformations, we measure the similarity between the test and training views with the cross-correlation of the distributions

$$(\phi^f_{\mathsf{test}^f} \star \phi^f_{\mathsf{train}_w^f})(w) \;=\; \int_{\mathcal{A}^f} \phi^f_{\mathsf{test}^f}\big(x\big) \quad \phi^f_{\mathsf{train}_v^f}\big(\,\mathrm{transform}_w(x)\big) \;\; \mathrm{d}x \;\; . \tag{2}$$

To efficiently obtain an approximate evaluation the integral of Eq. 2, we use Monte Carlo integration [36]. This involves drawing samples x_i ($\ell = 1 \dots L$) from the distribution ϕ^f_{test} (see Section 4.3), and computing the following sum:

$$(\phi^f_{\mathsf{test}^f} \star \phi^f_{\mathsf{train}_v^f})(w) \;\;\approx\;\; \frac{1}{L} \;\; \sum_i^L \;\; \phi^f_{\mathsf{train}_v^f}\big(\,\mathrm{transform}_w(x_i)\big) \;\; . \tag{3}$$

We can substitute the distribution $\phi^f_{\mathsf{train}_v^f}$ by its definition with KDE (as in Eq. 1). Assuming this distribution is represented by L' particles x_j (either the original image features extracted from the training images, or a resampled set of those as will be discussed in Section 4), we have

$$(\phi^f_{\mathsf{test}^f} \star \phi^f_{\mathsf{train}_v^f})(w) \;\;\approx\;\; \frac{1}{L.L'} \;\; \sum_i^L \sum_j^{L'} \;\; \mathrm{wt}(x_j) \; \mathcal{N}(x_i.pos; \mathrm{transform}_w(x_j.pos), \sigma_{pos}) \; \mathsf{K}^f(x_i.app; x_j.app) \, . \tag{4}$$

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.94)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

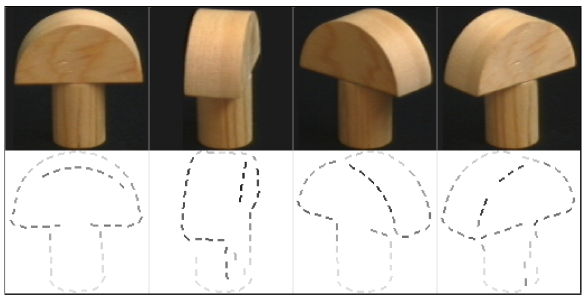


Figure 2: Visualization of weights assigned to edge points on a toy dataset (made of 18 images of the object rotating around one axis), computed using the training images alone (but no validation dataset, see text). Darker colors correspond to heavier weights. The parts looking similar in different views (e.g. the cylindrical base) receive lower weights, while the image features that can unambiguously determine a precise pose (e.g. non-silhouette edges) receive high weights.

Now, taking into account several types f of image features ($f = 1 \dots F$), the full similarity measure between two images finally uses the product over f of the expression above, which gives

$$\text{similarity}_{\mathsf{test},\mathsf{train}_v}(w) \;=\; \prod_f \; (\phi^f_{\mathsf{test}} \star \phi^f_{\mathsf{train}_v})(w) \quad . \tag{5}$$

We now have the core of the proposed method, with equations 4 and 5: we can easily evaluate the likelihood of observing, in the test image, the object under the viewpoint v and in-plane transformations w. The solution to the problem of object localization corresponds the maxima of Eq. 5, i.e.

$$(v^*, w^*) \;=\; \arg\max_{v,w} \Big(\text{similarity}_{\mathsf{test},\mathsf{train}_v}(w) \Big) . \tag{6}$$

Our algorithm to solve this maximiation problem is detailed in Section 4.2. It efficiently computes the values of the objective function over all image locations (in-plane translations), with a method similar to a Hough voting using samples drawn from our distributions of features.

2.4. Weighting of image features

We now present how to assign adequate weights to samples drawn from the trained model. The model of appearance presented in Sections 2.1 and 2.2 is merely a convenient way of representing the appearance of object categories. Since our goal is specifically to use this model to detect an object among clutter, and to determine its actual pose, we wish to give more weight its parts that are most informative to those tasks. As will be detailed in the Implementation section (Section 4), we choose to preselect samples offline from the trained model for efficiency. Therefore, the weights associated to these samples can also be computed in a pre-processing step, using the procedure described below.

Weighting training data in the context of object recognition is common among many existing methods [12, 37–39], where it has shown to increase performance significantly. In comparison to existing methods, our procedure is better suited to non-discriminative low-level image features, and does not rely on large amounts of training examples. It iteratively uses a validation test set to weight each feature relative to how informative it is to discriminate the appearance at a specific pose, versus other poses and against background clutter.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.95)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

The procedure is performed for each type f of image feature separately; we omit the superscripts f in the following paragraph to lighten the notations. We initially run the algorithm for detection and pose estimation (Section 2.3) with uniform weights on all image features of the training data. The idea is then to decrease the relative weight of those features that lead to incorrect results, from false positive detections (object identified in the background clutter) or from the recognition of incorrect poses (e.g. a car facing right identified as a car facing left). For each training view t (corresponding to a viewpoint v_t), we obtain some incorrect results $\{(v_n, w_n)\}_n$ $(n = 1 \dots N)$ to be used as negative examples (typically a pose estimate off by 20° or more, or an overlap of the detection bounding box less than 0.5 with the ground truth). We then update the weights of all image features x_i of the training view t according to a three step rule:

$$\begin{array}{lcl} \mathrm{wt'}(x_i) & = & 1 - \frac{1}{N} \sum_n \phi_{\mathsf{train}_{v_t}} \big(\mathrm{transform}_{w_n^{-1}}(x_i) \big) \\ \mathrm{wt}(x_i) & \leftarrow & \lambda \ \mathrm{wt}'(x_i) \ + \ (1 - \lambda) \ \mathrm{wt}(x_i) \\ \mathrm{wt}(x_i) & \leftarrow & \mathrm{wt}(x_i) \ / \ \sum_i \mathrm{wt}(x_i) \ . \end{array} \tag{7}$$

The first of these steps evaluates the contribution of the image feature x_i to the negative examples (incorrect results), by simply measuring how well that feature "matches" with the training view superimposed onto the test view (according to the in-plane transformations w_n). The weights are then updated (step 2, with learning rate $\lambda = 0.5$, typically), and normalized as to always sum to 1 (step 3). The effect of these steps is thus to actually decrease the relative weight of the features that lead to misdetections of misclassifications of the pose. The whole procedure is then be repeated iteratively: detection is performed, again, on the same validation dataset, but with the new weights for the model, which gives different negative examples, that are used with the three step rule to update the weights. As shown through our experiments, stable weights are usually reached within the order of 4–5 iterations (Section 5.2, Fig. 9).

Note that, if no validation test set is available, the weights can still be computed as described above by reusing, as validation test set, the training images themselves. When performing detection on the training images, the difficulty is then essentially to recognize the object in one viewpoint versus the other viewpoints (and not versus clutter). As a result, the weights then learned from negative results will help to differentiate each training viewpoint: higher weights are given to the image features that are very informative to a specific viewpoint (Fig. 2). This effect is similar to the one obtained in earlier work [1].

Finally, let us remark that the weighting scheme proposed here could be compared to the classical "term frequency – inverse document frequency" approach used in text mining, where high weights are assigned to words (image features, in our application) specific to a class of documents to retrieve (a specific viewpoint, here), relative to their likelihood of occurrence in general (in background clutter, in our case) [40].

3. Continuous pose estimation

The appearance model presented so far treats the different viewpoints provided in the training data independently, and performs a *coarse* pose estimation, or pose classification, by recognizing one of those discrete viewpoints. Our objective is now to provide a more accurate estimate of the pose, beyond the resolution of the training viewpoints. We first present a generative model capable of synthetizing the appearance of the learned object (or object category) at an arbitrary viewpoint, interpolating between the known views, then we show how to use it for a local optimization of initial (coarse) results.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.96)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

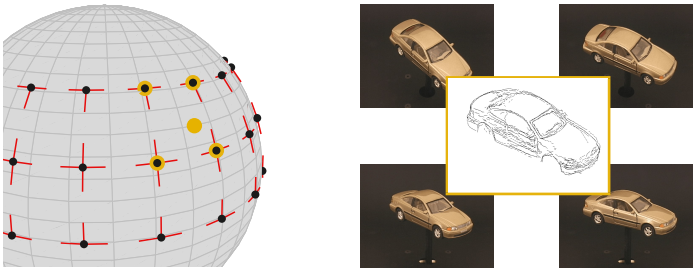


Figure 3: Generative model of appearance for novel viewpoints. The training viewpoints are typically regularly distributed on the viewing sphere (left, black dots) as here with the "Volvo car" dataset. Deformations between adjacent viewpoints (red segments) are detected with an optical flow algorithm, and allow interpolation of object appearance at a novel viewpoint (orange dot and center picture), by combining and deforming the image features of nearby viewpoints (orange circles and outer pictures).

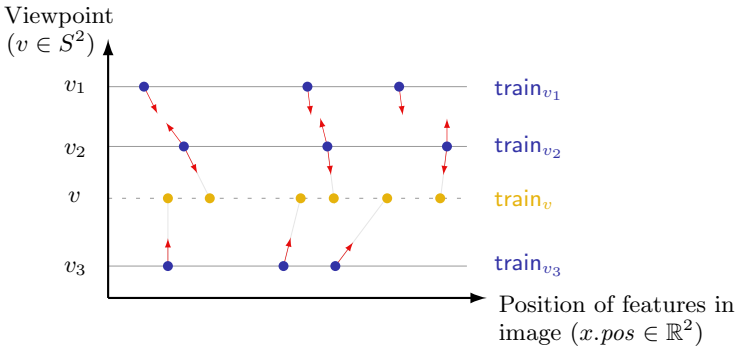


Figure 4: Schematic representation of the training data, and of the generative model for novel viewpoints. Image features (blue points) are available for some discrete training viewpoints v_t, and the deformations (translations in the image plane, red arrows) are detected between adjacent viewpoints during a preprocessing step. These translations can then be linearly interpolated to infer the appearance of the object at a novel viewpoint v (the set of image features train_v). The novel view includes the image features from several nearby training views, v_2 and v_3 here.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.97)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

3.1. Generative model of appearance for arbitrary viewpoints

The goal of our generative model is basically to fill in the gaps between the discrete training viewpoints. Although it is sometimes possible to establish explicit correspondences between image features of nearby training views, this approach could not be relied upon in general, as it does not generalize to dense or non-discriminative image features. Therefore, we chose instead to identify *dense* deformations between pairs of adjacent training views, using an optical flow algorithm. Those deformations are then combined and linearly interpolated to deform the image features of the training images into any arbitrary viewpoint (Fig. 3 and 4).

More precisely, we first define a function $\text{dist}(v, v')$ that measures the angular distance between two viewpoints on the viewing sphere. We define the set of all pairs of neighbouring training viewpoints $\mathcal{V} = \{(t, t') : \text{dist}(v_t, v_{t'}) < th\}$ (with a threshold of $th = 20°$ typically). During an offline training phase, an optical flow algorithm [41] is applied on all pairs of views $(t, t') \in \mathcal{V}$[1]. Each pair produces a dense flow map $\text{UV}_{t\to t'}(x)$ that corresponds, in our case, to the local deformation (translation in the image plane) undergone at an image location x when moving from viewpoint v_t to $v_{t'}$. We can now define our generative model noted train_v, which corresponds to the set of image features defining the appearance of the object at a novel viewpoint v, as the union of image features of nearby training views, translated appropriately using the precomputed deformations. Formally,

$$\mathsf{train}_v \;=\; \bigcup_{v_t:\text{dist}(v_t,v)<th} \text{deform}_{v_t\to v}\left(\mathsf{train}_{v_t}\right) . \tag{8}$$

The function $\text{deform}_{\text{v}_\text{t}\to\text{v}}$ adjusts the position of the image features of a training view v_t into the novel viewpoint v. It uses a linear combination of two precomputed deformations, in order to translate each image feature adequately. We denote these two deformations by the indices of the two viewpoints between which we computed them, and call them (t, t') and (t, t''). They are chosen from $\mathcal{V}$ so that the novel viewpoint can be reached (on the viewing sphere) by a positive linear combination of them. Therefore, $\exists\, \alpha, \beta \in \text{R}^+ : v = v_t + \alpha(v_{t'} - v_t) + \beta(v_{t''} - v_t)$. Practically, this means that the viewpoints v_t, $v_{t'}$ and $v_{t''}$ cannot be collinear on the viewing sphere. In the simple case where training viewpoints spaced on a grid (as in the experiments of Section 5), we simply choose $v_{t'}$ and $v_{t''}$ respectively along the changes in azimuth and elevation. It is now straightforward to define the function that combines the two deformations:

$$\begin{aligned} \text{deform}_{v_t\to v}\left(\mathsf{train}_{v_t}\right) = \quad & \{x'_i : x'_i.pos = x_i.pos \;\; + \alpha \;\; \text{UV}_{t\to t'}(x_i.pos) \\ & \qquad\qquad\qquad\qquad + \beta \;\; \text{UV}_{t\to t''}(x_i.pos) \\ & \text{and } x'_i.app = x_i.app \;, \quad \forall x_i \in \mathsf{train}_{v_t} \quad \} . \end{aligned} \tag{9}$$

The appearance of the image features is thus left unchanged, but their position in the image is modified using a linear combination of the deformations detected with optical flow. Using a parameterization of the viewpoint with euler angles as we do in our implementation (Section 4.4), this linear interpolation of image location with respect to angles is a simplistic approximation of the underlying transformations (3D rotation and projection onto the image place). This linear approximation however proved appropriate, since the deformations are detected between fairly close viewpoints (due to the limitations of the optical flow algorithm), and more complex interpolation schemes did not prove more effective in practice.

[1] When building a model of an object *category*, the deformations are detected using pairs of views of a single object instance at a time, since the detection of optical flow requires fairly similar images to succeed.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.98)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

Edge points			
$\mathcal{A}^{\text{edges}}$	$=$	$\mathbb{R}^2 \times S_1^+$	position, orientation
$\mathsf{K}^{\text{edges}}(x_1.app, x_2.app)$	$=$	$\text{VM}^+(x_1.app;\ x_2.app, \kappa)$	
	$=$	$C_1 \,.\, e^{\kappa \cos(x_1.app - x_2.app)}$	
Gradient points			
$\mathcal{A}^{\text{gradients}}$	$=$	$\mathbb{R}^2 \times S_1$	position, orientation
$\mathsf{K}^{\text{gradients}}(x_1.app, x_2.app)$	$\overset{1}{=}$	$C_2 \,.\, \text{VM}^+(x_1.app;\ x_2.app, \kappa)$	undirected
	$\overset{2}{=}$	$C_3 \,.\, \text{VM}\,(x_1.app;\ x_2.app, \kappa)$	directed

Table 1: Formal definition each type of image features used in our implementation. The notations VM and VM^+ denote von Mises disributions respectively on $S_1 = [0, 2\pi[$ and $S_1^+ = [0, \pi[$. C denotes a normalization constant.

3.2. Local optimization for the pose of initial detections

We use the algorithm of Section 2.3 to obtain initial detections and recognitions of training poses. Those are then used as starting points to run a local optimization, using the generative model described above, in order to refine and obtain a precise pose estimate. The objective function to maximize during this optimization is still the same as described in Section 2.3 (Eq. 5). The only difference now is that the similarity is measured between the test view and a *generated* view, at an arbitrary viewpoint. Since the appearance of a generated view varies smoothly across viewpoints, the value of the similarity measure (our objective function) is also guaranteed to be smooth in the neighbourhood of the optimum we are seeking. However, no assumption can be made about its convexity, and its complex definition (parameterized on the 6 dimensions of the viewpoint and in-plane transformations) makes the evaluation of its gradient expensive. Fortunately, the initial estimates used as starting points can be assumed to be close approximations of the global optimum. All those conditions motivated the use of a simple hill-climbing algorithm. We iteratively optimize pairs of dimensions at a time, namely the 2 viewpoint angles, the image location, then the scale and in-plane rotation. We empirically observed that a close approximation of the global optimum can be reached in this way after only a few iterations [2].

4. Implementation

This section presents details that are not specific to the method, but rather choices of implementation. Those specific choices discussed below refer to the implementation used throughout the evaluation of Section 5 and available on the author's website [42].

4.1. Application to different types of image features

We demonstrate the applicability of the method to two different types of image features: edges and intensity gradients extracted at a coarse scale.

Edge points

Image edges are widely used in the context of object recognition as they are effective and efficient representatives of shape (being rather sparse, compared to dense gradients). Using edges alone will also allow a fair comparison of our results with existing methods. We use the classical intensity-based Canny detector to extract edges from input images. The image features considered are then the

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.99)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

pixels belonging to the resulting binary edge map. We attach, to each of these *edge point* features, an appearance attribute corresponding to the local (tangent) orientation of the edge, defined on $S_1^+ = [0, \pi[$. The kernel associated with that attribute naturally uses a von Mises distribution (similar to a wrapped Normal distribution) on the half-circle (Table 1).

Note that our distance measure between edges could be compared to the directional chamfer distance [41, 43]. The approximation proposed in earlier work (discretization of orientations, approximation of edges by straight segments [41], etc.) can thus be seen as approximations of our more general formulation. Consequently and unsurprisingly, the directional chamfer distance was reported to perform similarly as our base method on the ETHZ shape dataset with hand-drawn examples [41]. This comparison is anecdotal since their exact performance numbers were not available. Moreover, our method includes numerous other improvements like the weights on the features or the learning of models from examples.

Gradient points

The goal of our *gradient* features is to represent regions in the image of slowly varying intensity, due e.g. to shading on smooth surfaces. It is easy to see how this information is complementary to the edges, which rather capture sharp transitions. We extract gradients by first convolving the image with derivative-of-Gaussian filters in horizontal and vertical orientations. Each pixel of the image with significant gradient magnitude (set by a fixed low threshold) is an image feature, which gets, as its appearance attributes, the orientation of the gradient (an angle in $[0, 2\pi[$). The extraction of gradients is performed at several coarse scales (typically, $\sigma = 2 \dots 5\,\text{px}$), and the gradient of largest magnitude is retained. We propose two versions of a kernel suited to the gradient points (Table 1), using the orientation in either an undirected or directed manner. In the *undirected* manner, the orientation of the gradients is compared only on the half-circle. Two horizontal gradients, from black to white and from black to white would thus be considered identical. In the *directed* manner, their orientation in that case is considered opposite. We compare both versions in our experiments.

4.2. Voting algorithm for object detection

As presented in Section 2.3, performing object detection amounts to identifying maxima of the similarity between the test view and one of the training view. We perform the initial detection using one single type of image features at a time. In practice indeed, in the problem of localization in an image, the meaningful optima of the full similarity function (using several types of image features, Eq. 5) will also correspond to local optima for each type features alone. For efficiency, we typically run this procedure using the (more sparse) *edge points*, and then compute the exact similarity scores with (possibly) additional features (Eq. 5), at those discrete values of (v, w) proposed by the voting algorithm. It is however also possible to use dense features alone (gradients for example) with this voting procedure, e.g. if the object does exhibit any meaningful edges, as demonstrated in Section 5.6.

From the definition of our similarity measure we show below that a procedure akin to a traditional Hough voting can approximate this value, which leads to Algorithm 1. On the one hand, considering a single type of features f, Eq. 4 and 5 specify how to measure the similarity between the test view and a training view v under the in-plane transformations w:

$$\text{similarity}_{\mathsf{test},\mathsf{train}_v}(w) \approx \frac{1}{L.L'} \sum_{i}^{L} \sum_{j}^{L'} \text{wt}(x_j)\ \mathcal{N}(x_i.pos; x'_j.pos, \sigma_{pos})\ \mathsf{K}^f(x_i.app; x_j.app)\,. \quad (10)$$

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.100)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

with samples x_i drawn from $\phi^f_{\mathsf{test}^f}$, x_j from $\phi^f_{\mathsf{train}^f_v}$, and $x'_j = \text{transform}_w(x_j)$. Let us consider a common 2D voting space $\mathcal{H}$ corresponding to image locations, containing discrete votes at locations $v_j.pos$ of respective weights $v_j.weight$. After convolving this voting space with an isotropic Gaussian kernel of bandwidth σ_{pos}, the value at a location l is given by:

$$\mathcal{H}(l) \;=\; \sum_j v_j.weight \;.\; \mathcal{N}\big(l; v_j.pos, \sigma_{pos}\big) \;. \tag{11}$$

One can now readily see that Eq. 10 and 11 can be made equivalent with votes in the Hough space such that $v_j.pos = (x'_j.pos - x_i.pos)$ and $v_j.weight = \text{wt}(x_j) \,.\, \mathsf{K}^f(x_i.app, x_j.app)$. Thus, by casting votes of such locations and weights, the values in the voting space after blurring will approximate our similarity measure for all the discrete image locations represented by the voting space, from which we can then trivially identify the local maxima. The complete algorithm is given in Algorithm 1. It iterates over discrete viewpoints, scales and in-planes rotations, then uses at each iteration the voting procedure to identify the best image location.

Algorithm 1 Voting algorithm for object detection, similar to a generalized Hough transform, which uses samples from our distributions of image features.

Input:

f	The type of image features to use for the initial detection.
test^f	Set of such image features of the test view.
$\mathsf{train}^f_{v_t}$	With $t = 1 \dots T$, sets of image features of the T training views.

Output:

$\mathcal{R} = \big\{(v_j, w_j)\big\}_j$	Candidate detections of training viewpoints, i.e. couples of viewpoint/in-plane transformations, local maxima of Eq. 2.

Procedure:

$\mathcal{R} \leftarrow \emptyset$

For each discrete training viewpoint $t = 1 \dots T$
  For each discrete step of in-plane rotation r
    For each discrete step of image scale s
      Initialize $\mathcal{H}$ empty 2D Hough accumulator corresponding to image locations
      For each $\ell = 1 \dots L$
        Select a sample x_1 from $\phi^f_{\mathsf{test}^f}$ and x_2 from $\phi^f_{\mathsf{train}^f_{v_t}}$
        Add a vote to $\mathcal{H}$ at location $x_2.pos - x_1.pos$
        of weight $\mathsf{K}^f(x_1.app, x_2.app) \,.\, \text{wt}(x_2)$
      Convolve ("blur") $\mathcal{H}$ with Gaussian kernel of size $(s.\sigma_{pos})$
      Keep each local maxima of $\mathcal{H}$: store its corresponding image location in w together with current rotation r and scale s, and $\mathcal{R} \leftarrow \mathcal{R} \cup (v_t, w)$

4.3. Building and sampling category models

In order to build a model of an object *category* from several instances, we first identify the discrete viewpoints provided in the training data, and at which the category model will be defined. For each viewpoint, we combine all instances defined at that viewpoint, by aligning the views and

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.101)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

simply merging their sets of features (see Fig. 8 for example). To align the views, we trivially translate and/or scale each example as it is added to the model, so as to maximize its similarity (Eq. 5) with the current model (Fig. 6, top row).

Using our distributions of features requires drawing samples from those. Sampling from distributions defined through KDE involves selecting a particle at random, then drawing a sample from its associated kernel. The set of particles that define category models is representative of the distribution of image features among the training examples, which is highly multimodal. If those examples are only roughly segmented and contain significant clutter, as in the "ETHZ Shape" dataset (see Fig. 6, top row), a large fraction of the particles will account for noise. They correspond to non-meaningful variations of appearance among the training examples that we wish *not* to capture. To address this specific concern, we propose an variant of the sampling procedure that focuses on the main modes of the distribution. This variant differs in the selection of a particle. Instead of choosing it uniformly at random, we select particles with a probability proportional to their likelihood under the distribution defined by the whole set of features. Formally, given the set of features $\mathsf{train}^f = \{x_i\}_{i=1}^{M_k}$, which define the distribution $\phi^f_{\mathsf{train}^f}$, we will select a particle x_i with a probability proportional to $\phi^f_{\mathsf{train}^f}(x_i)$. Similar procedures for drawing samples from the main modes of a distribution have been previously proposed in the literature, e.g. in [44] under the name of "2-level importance sampling". As a side note, formulated using importance sampling, the technique proposed above corresponds to using ϕ as the proposal distribution, in order to sample from a distribution ϕ' in which the probability densities would have been squared. Visual comparisons of sampling methods are provided in Fig. 6. Moreover, we empirically observed that, after selecting particles, drawing random samples from their associated kernels proved unnecessary or sometimes detrimental, unless using very large numbers of samples. We thus only use the subset of particles themselves as samples. For efficiency, we preselect this subset off-line as a preprocessing step. Those precomputed samples are thus readily available at test time, and this also allows precomputing their associated weights (Section 2.4). A complete overview of the different steps involved in the learning of a category model, then in its use for detection and pose estimation, is provided in Algorithm 2.

4.4. Software implementation

The manipulation of our low-level image features typically involves large numbers of very simple operations. These are excellent candidates for massively-parallel execution on a graphical processing unit (GPU). The provided software is implemented in Matlab and allows execution on either a CPU or a GPU. As a ballpark figure, on a typical consumer-level desktop computer, execution on a GPU is typically 20 times faster than execution on a CPU. Although some specific effort was spent adapting the algorithm for execution on a GPU, performance has not been our primary concern, and further improvements in performance are certainly possible. Existing work on the implementation of the Hough algorithm on GPUs [45–47] may be of interest in this context.

5. Experimental evaluation

All the contributions of this paper form together a single coherent framework. One of our goals is to demonstrate the versatility of the resulting method, which we therefore evaluate on a variety of tasks and datasets. We present them by order of relative complexity, starting with object detection, first learned from a clean shape template, then learned from images. We then consider the task of *coarse*, discrete pose estimation (or pose *classification*), i.e. the recognition of specific trained viewpoints. We finally consider *continuous* pose estimation. The task of pose estimation is viewed as the most complex task, as it does also involve the detection and recognition of the object within

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.102)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

Algorithm 2 Full algorithm for learning model of object category, and for detection followed by continuous pose estimation in a test image.

Training (off-line)

For each viewpoint
  Extract edge and gradient features from training images of the current viewpoint
  Align features of training images, as to maximize their similarity (Eq. 5)
  Merge aligned features of all those training images
  Pre-draw samples from resulting distribution, assign uniform weights

Extract edge and gradient features from validation images
Pre-draw samples from resulting distributions, assign uniform weights

For each iteration for learning weights
  Perform detection on validation images (Algorithm 1)
  Update weights using incorrect detections as negative examples (Eq. 7)

If training viewpoints are close enough for continuous pose estimation
  Detect deformations between neighbouring viewpoints with optical flow
  Store deformation of each pre-drawn sample from the training images

Testing (on-line)

Extract edge and gradient features from test image
Draw samples from resulting distribution, assign uniform weights
Perform detection, using edges only (Algorithm 1)
Compute full similarity scores of resulting detections, using edges and gradients (Eq. 5)
Return detections with highest scores

If training viewpoints are close enough for continuous pose estimation
  Consider the detection with the highest score
  For each iteration for optimizing the viewpoint
    Generate appearance of the model at a slightly perturbed viewpoint (Eq. 8)
    Compute similarity score between test image of generated viewpoint (Eq. 5)
    If similarity score improved **then** keep perturbed viewpoint
  Return the detection with the optimized viewpoint

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.103)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

					
Full proposed method (learned weights)	84.1/84.1	96.4/96.4	74.7/73.0	69.7/62.1	90.9/81.8
No weights	81.8/79.5	96.4/90.9	52.7/44.0	54.5/45.5	78.8/66.7
Contour networks, Ferrari, ECCV 2006 [48]	72.7/56.8	90.9/89.1	68.1/62.6	81.8/68.2	93.9/75.8
TPS-RPM, Ferrari, CVPR 2007 [49]	86.4/84.1	92.7/90.9	70.3/65.9	83.4/80.3	93.9/90.9
Ravishankar, ECCV 2008 *et al.* [50]	97.7/95.5	92.7/90.9	93.4/91.2	95.3/93.7	96.9/93.9

Table 2: ETHZ Shape dataset: detection with hand-drawn models. Weights on image features are represented on the first line; darker colors correspond to heavier weights. We report detection rates (in %) at 0.4/0.3 FPPI. We obtain performance in the order of state-of-the-art methods specifically designed for contour matching. We perform relatively poorly with giraffes and mugs though, which present more variety in aspect ratio in the test images.

clutter the image. To the extent possible, we reuse existing datasets, such as the "ETHZ shape" [48] and "3D Object" [27], considered as benchmark datasets. This allows direct comparison with recent and state-of-the-art methods on several of the tasks considered. Additionally, we present some of the unique capabilities of our method with a custom dataset of smooth and non-textured objects that can only be recognized from shading and homogeneous image regions, which we make possible through the use of coarse-scale image gradients as image features. All scripts for replicating the experiments of this paper are available, together with the code of the method, on the author's website [42]. Very few parameters need to be set within the method. A suitable bandwidth for the kernels (Eq. 1) is set as a fraction of the size of the object in the training images (for example, in the order of $\sigma_{pos} = 10$px for the ETHZ shape dataset), and the bandwith on the orientation of edges and gradients is set with $\kappa = 128$ (in a von Mises distribution, which would correspond to a standard deviation of $\sim 20°$ in a wrapped normal distribution). The effect of the other parameters is discussed below, notably the number of samples drawn from the distributions. We identify overlapping detections from the Hough voting algorithm as per a standard procedure, i.e. when their bounding box overlap exceeds 20%, then keep only the one of higher score. One practical effect is that, if two trained vewpoints are matched on a similar location in the test image, only the one with the highest similarity score is retained.

5.1. ETHZ Shape dataset: benchmark for shape detection, trained from a single or multiple examples

The *ETHZ Shape* dataset is a standard benchmark for object detection, which features five diverse classes (bottles, swans, mugs, giraffes and apple logos) in a total of 255 images collected from the web by Ferrari *et al.* [48]. It is considered very challenging because of intraclass shape variations, large scale variability and severe clutter. The goal of evaluating this dataset is to demonstrate that the proposed method achieves adequate performance of shape-based detection. Although we do achieve performance on this task on par with or superior to previously-proposed methods, our method was *not* specifically aimed at this task, and its many other capabilities will be demonstrated on other experiments presented below. The object classes of the ETHZ dataset are intrinsically defined by their shape, and we therefore focus on the use of image edges, as most competing methods do. We did not obtain significant differences in the results with other image features such as our coarse-scale gradients. We consider each object class separately, with a model (with

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.104)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

					
Full proposed method (proposed sampling, learned weights)	90.0/85.0	96.4/96.4	63.8/55.3	61.3/61.3	52.9/47.1
Proposed sampling, no weights	80.0/70.0	96.4/96.4	38.3/36.2	58.1/41.9	35.3/35.3
Random sampling, learned weights	25.0/25.0	53.6/53.6	12.8/14.9	6.5/ 9.7	23.5/23.5
Random sampling, no weights	20.0/20.0	75.0/71.4	17.0/12.8	29.0/22.6	23.5/23.5
HOG, Dalal, CVPR 2005 [7, 51]	85.0/ –	14.3/ –	34.0/ –	77.4/ –	67.7/ –
TPS-RPM*, Ferrari, CVPR 2007 [49]	83.2/77.7	81.6/79.9	44.5/40.0	80.0/75.1	70.5/63.2
kAS, Ferrari, PAMI 2008 [51]	60.0/50.0	92.9/92.9	51.1/49.0	77.4/67.8	52.4/47.1
M^2HT, Maji, CVPR 2009 [12]	95.0/95.0	96.4/92.9	89.6/89.6	96.7/93.6	88.2/88.2

Table 3: ETHZ Shape dataset: detection with models learned from images. The first line shows the training data, as all the training examples aligned and superimposed onto each other. We report detection rates (in %) at 0.4/0.3 FPPI. We obtain excellent performance on apple logos and bottles, but perform relatively poorly on the giraffes and the swans, for which the example images include lots of clutter. We do not reach the state-of-the-art performance of M^2HT, which includes an additional SVM-based classifier to validate candidate detections. * *The results of TPS-RPM are not directly comparable as they use a 5-fold cross validation.*

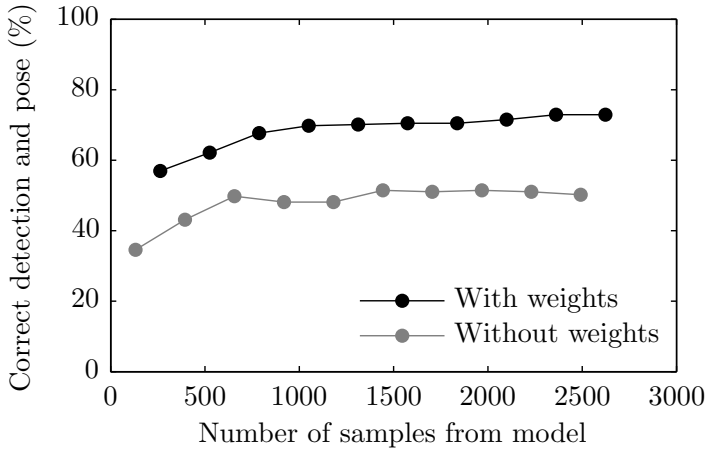


Figure 5: Influence of the number of samples used from the object model. We report the ratio of correct results (correct bounding box *and* correct estimated pose) on the 6^{th} car of the "3D Object" dataset. Performance degrades smoothly with smaller numbers of samples, which can be desirable for efficiency.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.105)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

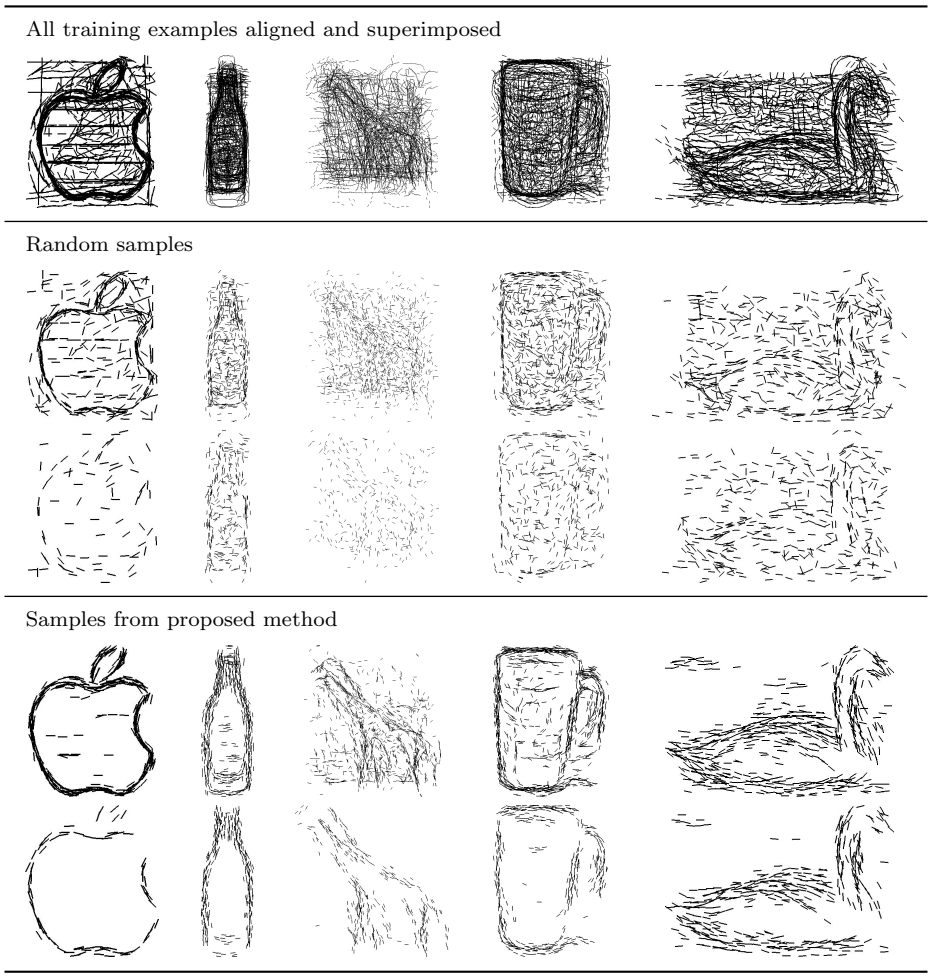


Figure 6: ETHZ Shape dataset: models learned from images, visualized as samples drawn from the distributions of features. We visualize two different amounts of samples for each sampling method (with equal amounts for the two methods). The proposed sampling scheme is able to recover very simple representations of the shapes with small number of samples, whereas a basic, random sampling includes unwanted samples corresponding to clutter in the training images. The model of the giraffe is noticeably worse than the other shapes, because of the large fraction of clutter in most of the training examples.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.106)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

a single viewpoint) trained for each of them independently. The common evaluation measure for this dataset is to plot detection rates (DR) versus the incidence of false positives (false positives per image, FPPI), while varying the detection threshold. Detection rates at a fixed FPPI of 0.3 are used for direct comparisons. Detections are counted as correct with a bounding box overlap of at least 20% with hand-drawn models, and 50% with models learned from images (again, as in existing work such as [51]). All parameters were kept identical for all object classes, except σ_{pos}, set from the size of the training template, as stated above.

The first setting we consider is the use of a single, hand-drawn model of each shape for training, as in [48]. The hand-drawn model is treated directly as an edge map, from which we pre-draw samples by selecting points along these edges, and of which we then learn weights. To allow a valid comparison with [18, 48], we use all 255 images as test set, and learn weights using incorrect detections (negative examples) in 20 random images collected from the web. We obtain the weights represented in Table 2. One can observe that long, uncharacteristic and easily matchable parts of the contours receive low weights, while high weights are assigned to salient parts with higher curvature, naturally less frequent among the random negative examples used to learn these weights. As expected, the detection results show that those weights significantly improve the results by decreasing the number of false positives (Table 2). While not surpassing the state of the art, we obtain remarkable performance, especially considering the fact that competing methods were specifically designed for the particular task of shape matching of contours, whereas our approach is a much more general one.

The second setting in which we evaluate this dataset involves learning the models from example images. We use the training and test splits of [51], i.e. the first half of the images of each class as the training set. We also use the rough presegmentation of these images provided as ground truth bounding boxes. Those images are aligned and set at a same scale (Section 4.3). We pre-draw samples from the model, of which we learn weights, using, as negative training images, images from the four other classes (as in [51]). The testing is performed on all other images of all classes. The models learned for each class are visualized in Fig. 6. The effect of the proposed sampling method (Section 4.3) versus a random sampling is quite dramatic. The proposed procedure concentrates on the main modes of the distributions, and provides reliable representations of the shape, even with limited numbers of samples. These "cleaner" models hide some undesirable variation from the training data, such as the water waves around the swans, or the inner texture within the apple logos.

We outperform a number of existing methods (Table 3). We do not reach the near-perfect results of M^2HT [12], which uses a discriminative classifier on top of their detections. Interestingly however, their detection algorithm alone achieved a rather low detection rate of only 60.9% at 1.0 FPPI, whereas our detector achieves 72.9% at 0.4 FPPI (averaged over the five classes). They also reported a notable improvement by performing detection at different aspect ratios, which we do not.

5.2. 3D Object dataset: multiview model, detection in clutter and coarse pose estimation

We now consider the "3D Object" dataset introduced by Savarese *et al.* [27]. We focus on the "car" object, as it is the most widely used, and gives us the most points of comparison with existing methods. The dataset features 10 different cars, each viewed under 24 viewpoints (8 azimuths and 3 elevations) and 3 scales. The task is both to detect the car among background clutter and to identify its azimuth angle (one of the 8 discrete values, i.e. whether it is view from the front, the left side, the 3/4 front/right side, etc). Pose estimation is limited to this coarse classification into the trained viewpoints, as these are too distant from each other to use our procedure for continuous

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.107)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

	Correct detections (AP)	**Correct poses (MPPE)**
Full proposed method (edges and directed gradients, learned weights)	92.5%	91.0%
Edges only, no weights	54.1%	85.2%
Edges only, learned weights	90.4%	92.6%
Arie, ICCV 2009 [52]	–	48.5%
Su, ICCV 2009 [53]	55.3%	67.0%
Liebelt, CVPR 2010 [54]	76.7%	70.0%
Payet, ICCV 2011 [55]	–	85.4%
Xiang, CVPR 2012 [56]	98.4%	93.4%

Table 4: 3D Object car dataset: detection and discrete pose estimation. The use of weights substantially improves the detection among clutter. We outperform most existing methods, although we do not reach the near-perfect performance of the "Aspect Layout Model" of Xiang *et al.* [56], which explicitly considers the 3D structure of the objects.

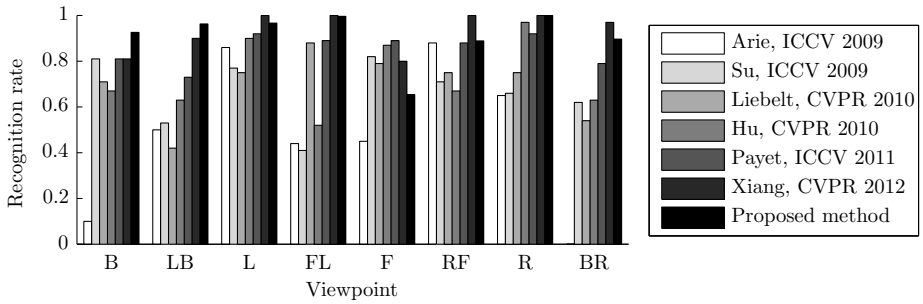


Figure 7: 3D Object Car dataset: classification rate of the different viewpoints. We clearly outperform most existing methods.

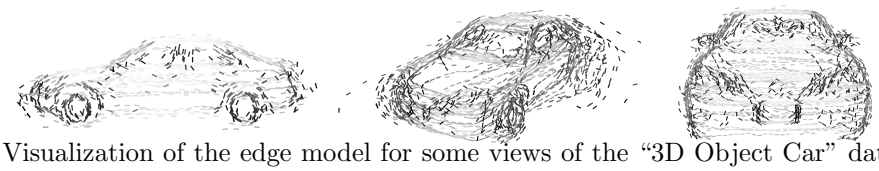


Figure 8: Visualization of the edge model for some views of the "3D Object Car" dataset. Darker colors correspond to heavier weights. Low weights are assigned to parts that can easily be matched to common background clutter (and lead to false positive detections), such as horizontal lines. More characteristic parts, such as the wheels in the side view, receive, on the opposite, high weights.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.108)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

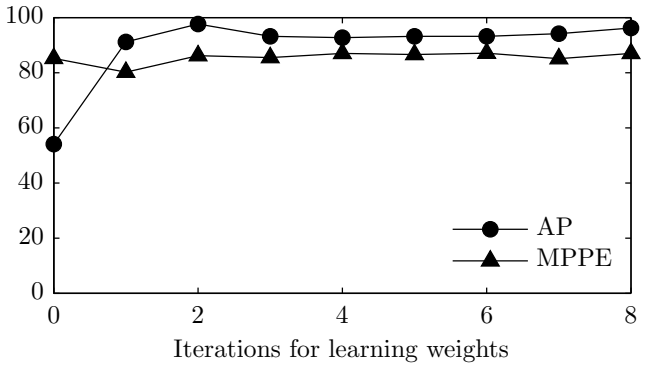


Figure 9: 3D Object Car dataset: evolution of performance for detection (AP) and pose estimation (MPPE) as a function of the number of iterations for learning the weights of the samples drawn from the distributions of features. At each iteration, weights are updated based on negative examples provided as incorrect detections in the training images themselves, then used in the manner of a validation dataset. Stable weights are reached with a small number of iterations.

pose estimation; finding dense correspondences between views of such complex objects would require viewpoints much closer than 45° apart.

We use similar conditions and evaluation criteria as [27]: the first 5 cars for training and the last 5 for testing. The training images are used both to build the model (with the provided ground truth segmentation), and then to learn weights by using the incorrect detections on them as negative examples (Section 2.4). Results are measured in terms of the rate of correct detections (average precision, or AP), defined by a bounding box overlap of 50%, and the ratio, among correct detections, of correct estimates of the azimuth angle (mean precision in pose estimation, or MPPE). As reported in Fig. 7 and Table 4, we outperform most existing methods evaluated on this dataset. The visualization of the weights learned for the image features (Fig. 8) provides some insight on their significant impact on performance. In the side view for example, the long horizontal lines, which are also frequent in background clutter, receive low weights. The wheels, on the opposite, are more characteristic and much better indicators of a car seen from the side, and thus receive higher weights. Interestingly, this distribution of weights is visually very similar to those obtained by Maji and Malik [12] with their own procedure, on side views of cars of the "UIUC car" dataset. We also observe that using our coarse-scale gradients as features, in addition to edges, brings a slight improvement. The difference is however marginal, as the appearance of the cars is already well defined by their shape and edges alone.

5.3. Tabletop dataset: multiview model, detection in clutter

We further evaluate our performance for object detection in clutter using the "tabletop" dataset of Sun *et al.* [57]. It features a total of 30 objects from 3 categories: computer mice, mugs and staplers. These object categories present more basic shapes than the cars in the "3D Object" dataset, which is a different challenge and provides complementary evaluation points. We use, as the training set, the part of the dataset with objects appearing on a turntable under known viewpoints ("Table-Top-Pose"; see Fig. 1). A model is learned for each object category. Testing is performed on scenes ("Table-Top-Local") containing one or several instances of the objects in a cluttered office environment; note that those experimental conditions are more challenging than

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.109)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

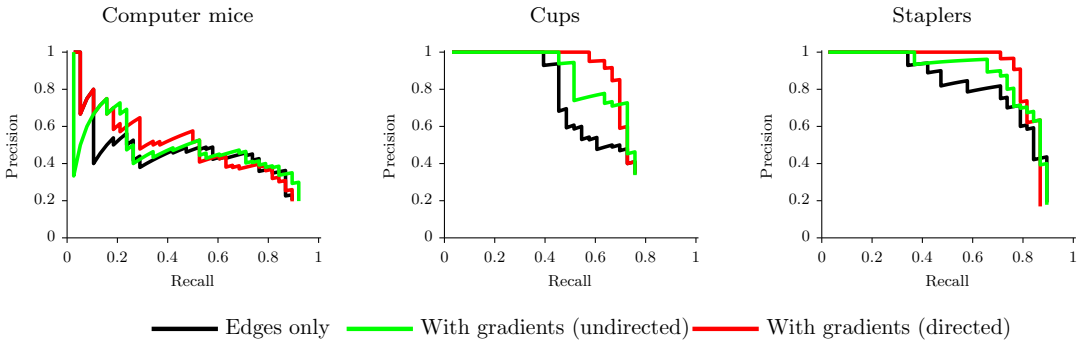


Figure 10: Results of detection on the tabletop dataset as precision/recall curves for each of the three object categories. The detection rate is significantly improved by using coarse-scale gradients in addition to edges, especially for the mugs, which present characteristic shading patterns captured by those additional features.

Figure 11: Sample detections of cups (left, center-left) and staplers (center-right, right) on the tabletop dataset (correct detections in green, incorrect ones in red).

Number of training views	15	30	40
Full proposed method (optimized viewpoint)	8.15°	**1.16°**	**0.80°**
Nearest neighbour detection only	8.63°	3.89°	3.00°
Torki and Elgammal [30]	5.47°	1.93°	1.84°
Teney and Piater, CRV 2013 [1]	**4.42°**	1.62°	1.49°

Table 5: Rotating car dataset: continuous pose estimation on a single instance (the first car). We report the mean error on the estimated azimuth angle, in degrees. We outperform existing methods on the two largest sizes of the training set; with smallest training set however, the viewpoints are often too distant to each other to reliably interpolate the appearance at intermediate viewpoints, and the optimization of the viewpoint is thus not as effective.

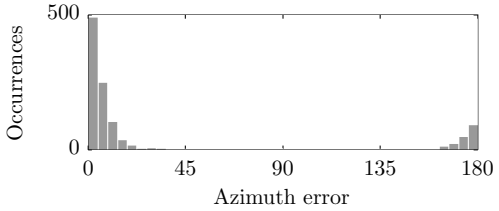


Figure 12: Rotating car dataset: distribution of error on estimated azimuth (in degrees) during experiments on multiple cars; a number of images yield an error of about 180°, due to ambiguous appearance of side views and front/rear views.

existing evaluations (e.g. in [55]) since those two parts of the dataset feature different imaging and lighting conditions. We perform detection in the test images of each object category separately, and we measure the detection rates with the standard criterion of 50% bounding box overlap. We report results in Fig. 10 as precision/recall curves. The use of coarse-scale gradients brings here a significant improvement, in particular on cups, the shape of which produces very characteristic shading patterns. The improvement is marginal for the computer mice: the different instances are very diverse in shape, and observed under fixed lighting conditions in the training images that produce specular highlights, which do not appear in the testing images. The simple gradients are obviously not robust to such variations by themselves, but we believe that they would show a better advantage if the training images presented more varied lighting conditions, although this could unfortunately not be tested with this dataset.

5.4. EPFL Rotating cars dataset: continuous pose estimation

We now evaluate the unique capability to perform *continuous* pose estimation within our appearance-based method. Few other methods have tackled this problem, especially at the level of object *categories*, which explains the limited choice of suitable datasets. The most appropriate, in our view, is the "Multiview car" dataset introduced by Ozuysal *et al.* [58]. It includes about 2000 images of 20 very different cars filmed on rotating stands at a motor show. The dataset is very challenging due to changing lighting conditions, high intraclass variability in shape, appearance and texture, and highly similar views (symmetrical side views, similar front and rear views) which are sometimes hard to differentiate even for a human. The dataset was used in [58] for pose classification in 16 discrete bins, and in [30] for continuous pose estimation. We first evaluate our method, as in [30],

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.111)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

	Median	Mean 90%ile	Mean	Error $< 22.5°$	Error $< 45°$
Full proposed method (optimized viewpoint, learned weights)	**5.2°**	**18.7°**	34.7°	**80.3%**	**82.1%**
Nearest neighbour detection only, no weights	7.6°	24.7°	39.8°	71.6%	76.3%
Nearest neighbour detection only, learned weights	5.7°	19.1°	35.0°	80.2%	82.1%
Ozuysal *et al.* [58]	–	–	46.5°	41.7%	71.2%
Glasner *et al.* [59]	24.83°	–	–	–	–
Torki and Elgammal [30]	11.3°	19.4°	**34.0°**	70.3%	80.7%
Teney and Piater, CRV 2013 [1]	5.8°	23.7°	39.0°	78.1%	79.7%

Table 6: Rotating car dataset: continuous pose estimation at the category level. Instances 1–10 are used for training (first row of pictures) and 11–20 for testing (second row of pictures). We outperform existing methods. Note however that the precision of the best methods reaches the accuracy and level of imprecision (estimated around $3-4°$) in the ground truth annotations, which explains why no further improvements can be made, especially by our optimization of the viewpoint.

	Detection rate (AP)	Azimuth error $< 10°$	Mean azimuth error
Full proposed method (optimized viewpoint)	83.3% (40/48)	70.0%	3.8°
Nearest neighbour detection only	83.3% (40/48)	70.0%	4.0°
Zia *et al.* [60] (with hand-made CAD models)	93.8%(45/48)	73.3%	3.8°

Table 7: Detection and continuous pose estimation, using the model learned from rotating cars (instances 1–10, as in Table 6), tested on images from the "3D Object" dataset (instance 6). The model is able to detect and estimate the orientation of the car accurately, despite challenging differences in imaging conditions, in scale and in object appearance between the two datasets. We use the same metrics as [60]: the rate of correct azimuths is measured on correct detections, and the mean error is measured on those correct azimuths. Incorrect azimuths often are off by about 180°. The results of Zia *et al.* [60] are included for reference only: they rely on hand-built CAD models, whereas our method is purely appearance-based and trained on example images.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.112)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

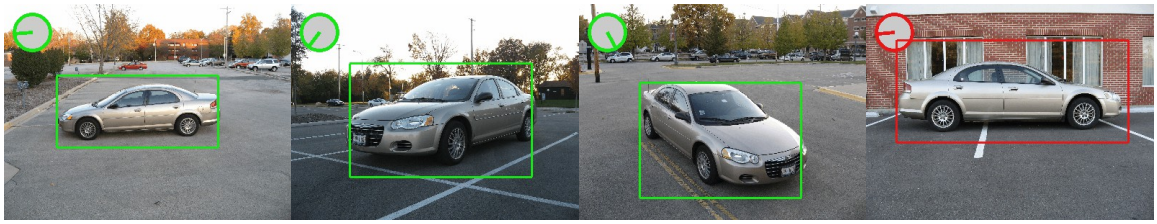


Figure 13: Samples results of continuous pose estimation on the "3D Object" dataset using the model learned from rotating cars. Boxes indicate the localization of the car as identified by our system, and the roses in the upper-left corners indicate the orientation of the front of the car as seen from the top (as in [58]).

on the first car of the dataset, training a model on this single specific car. We select 15, 30 or 40 equally-spaced images of the sequence as training images, and use all other images (spaced about 4° apart) for testing. We obtain superior results to [30] (Table 5). We then perform experiments at the *category* level, in conditions similar to those used in [30]. The first 10 cars of the dataset are used for training, and the other 10 for testing. Again, we obtain performance superior to all published results to our knowledge (Table 6). As highlighted in Fig. 12, the remaining errors in pose estimation correspond to an error of about 180°. This is caused by the symmetric aspects of some cars in the side views, and confusion between front- and rear-facing views.

We further evaluated the generalization capabilities of the model learned from this dataset. We thus use this model, trained from the 10 first rotating cars, for testing on the "3D Object" dataset (see Section 5.2 above). This is a challenging task, as those two datasets present very different conditions in terms of imaging conditions, scale, background clutter, etc. We do the testing specifically on the sixth car of the dataset, the exact pose of which was annotated by Zia *et al.* [60] by manually fitting 3D models to the images. These annotations are used as ground truth to measure the accuracy of the azimuth angle estimated by our method for continuous pose estimation. We obtain excellent results (Fig. 13), close to the accuracy obtained by the complex method of Zia *et al.* [60], which basically aligns 3D CAD models of cars with the images, compared to our more general appearance-based procedure.

5.5. Volvo car: continuous 3D pose estimation and synthesis of novel views

We further evaluate our method for continuous pose estimation, this time with a model spanning both dimensions of the viewing sphere around the object, as opposed to the single degree of freedom (azimuth angle) of the rotating cars presented above. The choice of datasets for this task that allow comparison with existing methods is limited, here again. We use the "3D pose Volvo car" of Viksten *et al.* [61, 62] (Fig. 14). This allows a comparison with a classical method [61] that uses discriminative image descriptors with a voting and averaging scheme, which is the classical approach for robust 3D pose estimation (with the disadvantage of being limited to specific object instances). The dataset features a toy car viewed under regular increments of azimuth and elevation angles. We consider two training/test splits: a small and a large training set, with views spaced respectively 20° and 10° apart (on both azimuth and elevation angles), and exactly one test view between each pair of training view, i.e. as a grid on the viewing sphere (as in [61]). In both cases, we obtain results significantly superior to [61] in terms of accuracy (Table 8). The smaller training set is more challenging for detecting deformations between views, and seemed to reach the limits of the optical flow algorithm we use to detect deformations between neighbouring views. The dataset also

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.113)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

Spacing between training views		**20°**	**10°**
Full proposed method (optimized viewpoint)	Azimuth	27.22°(1.67°)	**0.84**°(1.11°)
	Elevation	**2.65**°(1.11°)	**0.86**°(0.56°)
Nearest neighbour detection only	Azimuth	35.56°(10.00°)	5.00°(5.00°)
	Elevation	10.00°(10.00°)	5.14°(5.00°)
Johansson *et al.* [61]	Azimuth	**4.21**°	1.25°
	Elevation	2.66°	1.06°

Table 8: Continuous pose estimation on the Volvo car. We report the mean (median) error of azimuth/elevation angles, in degrees. The large mean error in azimuth comes from a single misclassified test image, as attested by the small median error. We clearly outperform the classical method of Johansson *et al.* based on discriminative feature descriptors and an averaging scheme in pose space.

Figure 14: Training images of the Volvo car with views spaced 20° apart.

allows a good visualization of the capabilities of our generative model, by varying continuously the viewpoint around the object. The effect, unfortunately hard to convey in static images (Fig. 15), is a vivid impression of manipulating a 3D model of the object – although there is no underlying explicit representation of the 3D shape. Videos and an interactive viewing tool are available on the author's website [42].

5.6. Non-textured objects

We finally demonstrate the interest of using coarse-scale gradients with a new dataset featuring non-textured objects. These toy objects, made of plastic, feature basic shapes with few internal edges (Fig. 16). This lack of distinctive visual characteristics actually makes them difficult to identify among clutter, and the absence of texture renders the estimation of their pose problematic. For example, considering the knife, one cannot differentiate the (round) handle from the (flat) blade, observing edges and silhouette alone. We made this new dataset available on the author's website [42]. It comprises examples images of each object with segmentation and pose annotations (used for training), plus a series of test images of cluttered scenes feature these objects, also with ground truth segmentations and annotations (used for evaluation). Results of detection are counted as correct when the overlap of bounding boxes with the ground truth exceeds 50% *and* when the estimated pose is correct (error on viewpoint angles smaller than 20°). Unsurprisingly, most objects are detected very poorly using edges alone (with our method; other existing contour-based recognition methods are expected to work as poorly). Our full method however, which uses coarse-scale gradients in the measure of similarity between the detections and the training examples, is able to differentiate

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.114)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

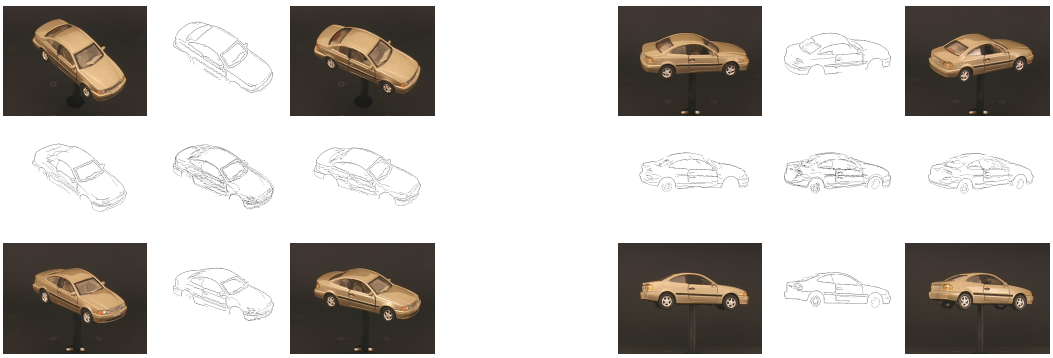


Figure 15: Demonstration of our generative model, with training views spaced 20° apart in azimuth and elevation angles, and the edge-based appearance of the object generated at intermediate, unseen viewpoints.

between similar-looking poses, and achieves far superior detection rates (Fig. 16). Most remaining incorrect detections are due to clutter and confusion from the similar appearance of these simple objects. We also tested the detection of those objects using gradient features alones, without edges. This did not prove effective in practice, since their appearance, defined by these gradients, is very simple and easily confused with the background or other objects. The knife for example, just corresponds to a region without gradients (the flat blade) and a part with gradients oriented orthogonally to the knife's length (the round handle). Such a description is complementary to the silhouette represented by edges, but is not informative enough by itself to localize such an object among clutter.

6. Discussion and conclusions

We introduced a representation of 2D appearance as distributions of low-level, fine-grained image features. We used this representation to build multiview models of object categories. Those models encode the appearance of objects at a number of discrete viewpoints, and, in addition, how these viewpoints deform into one another as the viewpoint continuously varies. Those deformations between neighbouring viewpoints are detected with an optical flow algorithm, and encoded as translations of individual image features with respect to viewpoint changes. We provide a way to measure the similarity between an arbitrary test image and an object model at a specific viewpoint. We use this measure of similarity to perform a number of tasks: detection and localization in cluttered images (identifying the local maxima of the similarity measure with respect to locations in the test image), discrete pose estimation (identifying the learned viewpoint with the highest similarity measure with the test image) and continuous pose estimation (identifying the maxima of the similarity measure as the viewpoint *continuously* varies). In contrast with common practice, we address and evaluate a number of related tasks with a single approach. This is reflected in our experimental evaluation, which includes extensive testing on a number of very different benchmark datasets, which are seldom considered together. We demonstrate performance on the "ETHZ Shape" dataset for shape matching and detection in clutter of categories well above baseline methods, on par with a number of more task-specific methods. We also obtain remarkable performance on the recognition of more complex objects, notably the cars of the "3D Object" dataset, with detection rates of 92.5%

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.115)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

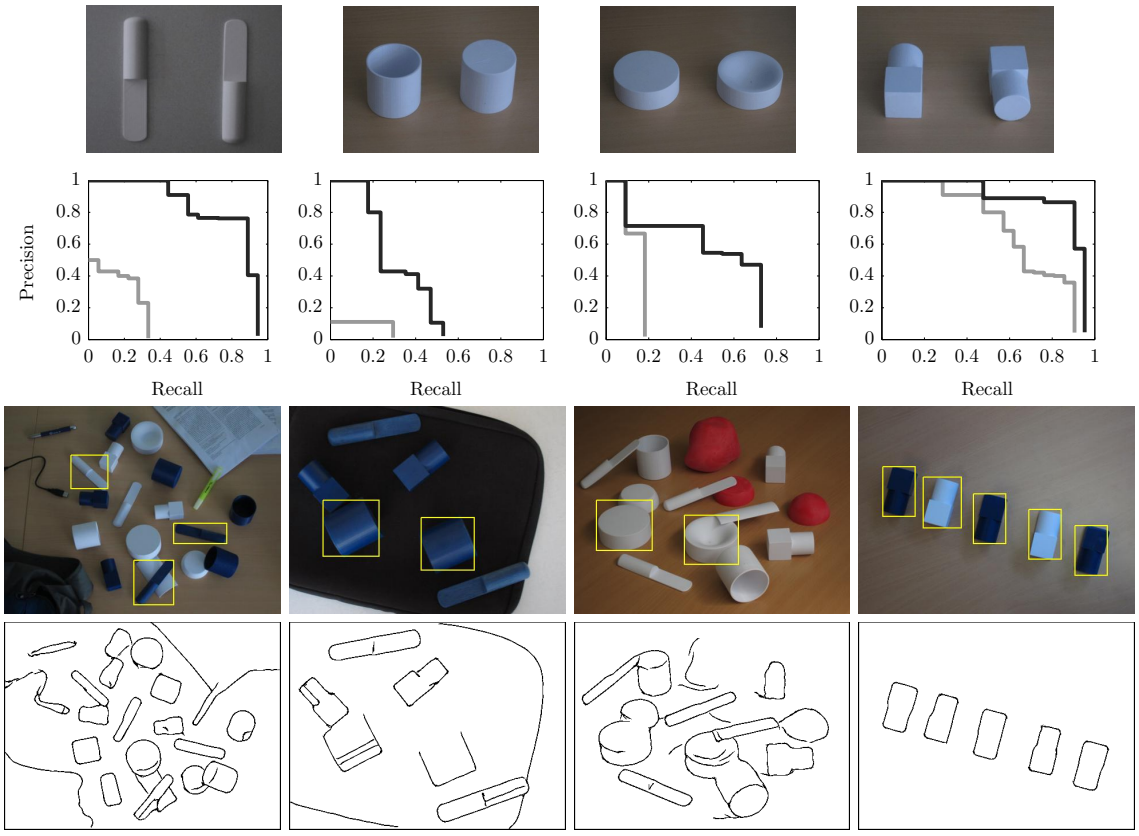


Figure 16: New dataset featuring non-textured objects with few distinctive characteristics. Sample training images (first row) of the objects, the "knife" (round handle, flat blade), the "cup", the "ashtray" (both hollow on one side only) and the "peg" (round on one side, square on the other). Results of detection and pose estimation on a total of 28 scenes are reported as interpolated precision/recall curves, using edges only (gray) and in conjunction with coarse-scale gradients (black). Sample scenes (bottom rows, with bounding boxes of highest-scored detections) show that edges provide only ambiguous information to determine the pose of the objects.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.116)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

and an accuracy in pose estimation of 91%. For the task of continuous pose estimation, we obtain results superior to the state-of-the-art on the "rotating cars" dataset.

The limitations of our appearance model lie mostly in the representation of object categories. The distrbution of image features are representative of the occurrence of features among the training examples, but they do not encode the co-occurrence of these features. The resulting model can thus represent all combinations of variations present in the examples. A model learned from images of cars and giraffes would not only represent those two types of objects, but also anything looking part-car and part-giraffe. This may be seen as a strength, as few examples can suffice to represent wide variations of overall appearance. However, this also means that the overall procedure will practically be most effective with training examples sharing strong visual characteristics, and not with categories defined semantically or including instances looking vastly different. This representation of appearance thus also assumes fairly rigid objects (although we still obtained good performance on shape matching of the ETHZ classes). Complex deformable objects would probably be better handled by part-based models (e.g. [17, 63]). We believe that this limitation was probably masked by the relative simplicity of the objects in the available datasets. Let us note however that the proposed representation as distributions of features could serve as a building block of part-based models.

The importance of shape and structure in the model leads to another limitation, in the context of object recognition in complex scenes. As opposed to, e.g. the classical "bag of visual words" approach, our model does not encode contextual clues of the scene. For example, blue color and clouds in the background of an image may be indicative of the presence of an airplane. Such information is however not encoded within our model, aimed at individual object recognition. This information could be taken into account at another, higher level, dealing for overall scene understanding.

All limitations discussed above lead to potential avenues for further developments. In addition, on the task of continuous pose estimation, one could explore alternative optimization algorithms to use with our generative model. Improvements in efficiency at this level could render the model suitable for continuous pose *tracking*, thereby widening its range of applicability even further. The detections of the deformations between the trained viewpoints, which currently uses a standard algorithm to detect optical flow, could also be improved, be made applicable to more distant viewpoints and to other types of training data, e.g. videos of the object.

Funding

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience. Damien Teney is supported by a research fellowship of the Belgian National Fund for Scientific Research (FNRS).

References

[1] D. Teney, J. Piater, Continuous Pose Estimation in 2D Images at Instance and Category Levels, in: Computer and Robot Vision, 2013.

[2] D. Teney, J. Piater, Modeling Pose/Appearance Relations for Improved Object Localization and Pose Estimation in 2D images, in: 6th Iberian Conference on Pattern Recognition and Image Analysis, vol. 7887 of *LNCS*, 59–68, 2013.

[3] F. Rothganger, S. Lazebnik, C. Schmid, J. Ponce, 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints, International Journal of Computer Vision (IJCV) 66 (3) (2006) 231–259.

[4] D. Hoiem, C. Rother, J. M. Winn, 3D LayoutCRF for Multi-View Object Class Recognition and Segmentation, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[5] P. Yan, S. M. Khan, M. Shah, 3D Model based Object Class Detection in An Arbitrary View, in: IEEE International Conference on Computer Vision (ICCV), 2007.

[6] D. Glasner, M. Galun, S. Alperta, R. Basri, G. Shakhnarovich, Viewpoint-Aware Object Detection and Continuous Pose Estimation, Image and Vision Computing .

[7] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 886–893, 2005.

[8] C. H. Lampert, M. B. Blaschko, T. Hofmann, Beyond sliding windows: Object localization by efficient subwindow search, in: In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR, 1–8, 2008.

[9] D.H., Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition 13 (2) (1981) 111 – 122, ISSN 0031-3203.

[10] B. Leibe, A. Leonardis, B. Schiele, An Implicit Shape Model for Combined Object Categorization and Segmentation, in: Towards Category-Level Object Recognition, 496–510, 2006.

[11] A. Lehmann, B. Leibe, , L. V. Gool, PRISM: PRincipled Implicit Shape Model, in: British Machine Vision Conference (BMVC), 2009.

[12] S. Maji, J. Malik, Object detection using a max-margin Hough transform, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[13] S. Ekvall, F. Hoffmann, D. Kragic, Object Recognition and Pose Estimation for Robotic Manipulation using Color Cooccurrence Histograms, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.

[14] P. Mittrapiyanuruk, G. N. DeSouza, A. C. Kak, Calculating the 3D-pose of Rigid-objects using Active Appearance Models, in: IEEE International Conference on Robotics and Automation (ICRA), 2004.

[15] A. R. Pope, D. G. Lowe, Probabilistic Models of Appearance for 3D Object Recognition, International Journal of Computer Vision (IJCV) .

[16] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision (IJCV) 60 (2) (2004) 91–110.

[17] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part-Based Models, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (9) (2010) 1627–1645, ISSN 0162-8828.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.118)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

[18] V. Ferrari, F. Jurie, C. Schmid, From Images to Shape Models for Object Detection, International Journal of Computer Vision (IJCV) 87 (3) (2010) 284–303.

[19] A. Opelt, A. Pinz, A. Zisserman, Learning an Alphabet of Shape and Appearance for Multi-class Object Detection, International Journal of Computer Vision (IJCV) .

[20] M. Martinez Torres, A. Collet Romea, S. Srinivasa, MOPED: A Scalable and Low Latency Object Recognition and Pose Estimation System, in: IEEE International Conference on Robotics and Automation (ICRA), 2010.

[21] F. Viksten, R. Soderberg, K. Nordberg, C. Perwass, Increasing pose estimation performance using multi-cue integration, in: IEEE International Conference on Robotics and Automation (ICRA), 2006.

[22] H. Schneiderman, T. Kanade, A statistical method for 3D object detection applied to faces and cars, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, ISSN 1063-6919, 746–751, 2000.

[23] A. Torralba, K. Murphy, W. Freeman, Sharing features: efficient boosting procedures for multiclass object detection, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, ISSN 1063-6919, II–762–II–769, 2004.

[24] M. Weber, W. Einhauser, M. Welling, P. Perona, Viewpoint-invariant learning and detection of human heads, in: IEEE International Conference on Automatic Face and Gesture Recognition, 20–27, 2000.

[25] D. Teney, J. Piater, Generalized Exemplar-Based Full Pose Estimation from 2D Images without Correspondences, in: Digital Image Computing: Techniques and Applications, 2012.

[26] A. Kushal, C. Schmid, J. Ponce, Flexible Object Models for Category-Level 3D Object Recognition, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 2007.

[27] S. Savarese, L. Fei-Fei, 3D generic object categorization, localization and pose estimation, in: IEEE International Conference on Computer Vision (ICCV), ISSN 1550-5499, 2007.

[28] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, L. Van Gool, Towards Multi-View Object Class Detection, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[29] D. Baltieri, R. Vezzani, R. Cucchiara, People orientation recognition by mixtures of wrapped distributions on random trees, in: IEEE European Conference on Computer Vision (ECCV), ISBN 978-3-642-33714-7, 270–283, 2012.

[30] M. Torki, A. M. Elgammal, Regression from local features for viewpoint and pose estimation, in: IEEE International Conference on Computer Vision (ICCV), 2011.

[31] S. Avidan, A. Shashua, Novel view synthesis in tensor space, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 1997.

[32] S. E. Chen, L. Williams, View interpolation for image synthesis, in: SIGGRAPH, 1993.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.119)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

[33] S. M. Seitz, C. R. Dyer, Toward Image-Based Scene Representation Using View Morphing, in: International Conference on Pattern Recognition, 84–89, 1996.

[34] S. Savarese, L. Fei-Fei, View synthesis for recognizing unseen poses of object classes, in: IEEE European Conference on Computer Vision (ECCV), 2008.

[35] M. Sun, H. Su, S. Savarese, L. Fei-Fei, A multi-view probabilistic model for 3D object classes, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 2009.

[36] R. E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, Acta Numerica 7 (1998) 1–49, ISSN 1474-0508.

[37] A. Frome, Y. Singer, F. Sha, J. Malik, Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification, in: IEEE International Conference on Computer Vision (ICCV), 2007.

[38] C. Gu, J. J. Lim, P. Arbelaez, J. Malik, Recognition using regions, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[39] P. Yarlagadda, B. Ommer, From Meaningful Contours to Discriminative Object Shape, in: IEEE European Conference on Computer Vision (ECCV), 2012.

[40] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, New York, NY, USA, ISBN 0521865719, 9780521865715, 2008.

[41] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, R. Chellappa, Fast directional chamfer matching, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 1696–1703, 2010.

[42] D. Teney, Personal website, `http://montefiore.ulg.ac.be/~dteney/cviu.htm`, 2013.

[43] J. Shotton, A. Blake, R. Cipolla, Multiscale Categorical Object Recognition Using Contour Fragments, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 30 (7) (2008) 1270–1281.

[44] R. Detry, N. Pugeault, J. Piater, A Probabilistic Framework for 3D Visual Object Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 31 (10) (2009) 1790–1803.

[45] J. Gómez-Luna, J. M. González-Linares, J. I. Benavides, E. L. Zapata, N. G. Mata, Load Balancing versus Occupancy Maximization on Graphics Processing Units: The Generalized Hough Transform as a Case Study, IJHPCA 25 (2) (2011) 205–222.

[46] N. Jotwani, S. Sah, A Fast and Accurate GHT Implementation on CUDA, in: International Conference on Meta Computing, 2011.

[47] G.-J. van den Braak, C. Nugteren, B. Mesman, H. Corporaal, Fast Hough Transform on GPUs: Exploration of Algorithm Trade-Offs, in: ACIVS, 611–622, 2011.

[48] V. Ferrari, T. Tuytelaars, L. Van Gool, Object detection by contour segment networks, in: IEEE European Conference on Computer Vision (ECCV), ISBN 3-540-33836-5, 978-3-540-33836-9, 14–28, 2006.

[49] V. Ferrari, F. Jurie, C. Schmid, Accurate Object Detection with Deformable Shape Models Learnt from Images, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 1–8, 2007.

[50] S. Ravishankar, A. Jain, A. Mittal, Multi-stage contour based detection of deformable objects, in: IEEE European Conference on Computer Vision (ECCV), 483–496, 2008.

[51] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, Groups of Adjacent Contour Segments for Object Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 30 (1) (2008) 36–51.

[52] M. Arie-Nachimson, R. Basri, Constructing Implicit 3D Shape Models for Pose Estimation, in: IEEE International Conference on Computer Vision (ICCV), 2009.

[53] H. Su, M. Sun, L. Fei-Fei, S. Savarese, Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories., in: IEEE International Conference on Computer Vision (ICCV), 2009.

[54] J. Liebelt, C. Schmid, Multi-view object class detection with a 3D geometric model, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 1688–1695, 2010.

[55] N. Payet, S. Todorovic, From contours to 3D object detection and pose estimation (2011) 983–990.

[56] Y. Xiang, S. Savarese, Estimating the Aspect Layout of Object Categories, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISBN 978-1-4673-1226-4, 3410–3417, 2012.

[57] M. Sun, G. Bradski, B.-X. Xu, S. Savarese, Depth-encoded hough voting for joint object detection and shape recovery, in: IEEE European Conference on Computer Vision (ECCV), 658–671, 2010.

[58] M. Ozuysal, V. Lepetit, P. Fua, Pose estimation for category specific multiview object localization, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 2009.

[59] D. Glasner, M. Galun, S. Alpert, R. Basri, G. Shakhnarovich, Viewpoint-aware object detection and pose estimation, in: IEEE International Conference on Computer Vision (ICCV), ISSN 1550-5499, 1275–1282, 2011.

[60] Z. Zia, M. Stark, K. Schindler, B. Schiele, Revisiting 3D Geometric Models for Accurate Object Shape and Pose, in: IEEE International Workshop on 3D Representation and Recognition (3DRR), 2011.

[61] B. Johansson, A. Moe, Patch-duplets for object recognition and pose estimation, in: Computer and Robot Vision, 2005.

[62] F. Vikstén, P.-E. Forssén, B. Johansson, A. Moe, Comparison of local image descriptors for full 6 degree-of-freedom pose estimation, in: IEEE International Conference on Robotics and Automation (ICRA), 2779–2786, 2009.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.121)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

[63] B. Leibe, A. Leonardis, B. Schiele, Robust Object Detection with Interleaved Categorization and Segmentation, International Journal of Computer Vision 77 (1-3) (2008) 259–289, ISSN 0920-5691.

This article is part of Chapter 8: *Unified presentation and evaluation of our contributions on exemplar-based recognition* (p.122)
It has been submitted to: *Computer Vision and Image Understanding (CVIU)*

Chapter 9

Application to the recognition of a robotic arm

This chapter presents preliminary results on a practical application of our object recognition method. In the context of robotic manipulation, we apply it to the recognition, not of the manipulated object, but of the manipulator itself, i.e. the robot arm. This is a frequent requirement when visually monitoring a scene involving robotic manipulation of objects. The goal is, on the one hand, to assess or confirm the current configuration of the robotic arm (e.g. in terms of joint angles, or position of the end effector in the 3D space), and, on the other hand, to identify and segment, from the image of the scene, parts belonging to the robots itself versus elements of the scene, for example the manipulated object. The current practice, in order to make the use of this visual information easier, is to use indirect recognition methods, by attaching fiducial markers to the robot ([59] for example), which are easy to detect reliably. Such indirect solutions are however not satisfactory in a general case. The markers also need to be constantly visible to the camera, which is practically constraining. The particular robot arm used in our lab at the University of Innsbruck is a Kuka LWR (Fig. 9.1). It is made of five articulated links of simple smooth shapes. They present few distinctive visual characteristics and no texture, which makes our recognition method an excellent candidate for identifying it in images.

The method for recognizing the complete robot arm in an image is described in a short paper, included below, which was presented at the *Interactive Perception Workshop* at the 2013 *International Conference on Robotics and Automation* (ICRA). The method operates in two steps. First, it recognizes the individual links of the robot arm, as it would do for any other object. It then enforces the constraints between adjacent links, corresponding to the physical kinematic constraints of the joints of the arm. This results in the identification of a plausible overall configuration of all the links of the arm (Fig. 9.2).

The training data used for building the appearance model of the links, i.e. example images of these links under various viewpoints, was initially generated with computer graphics by rendering a 3D model of these links. This option was chosen to make

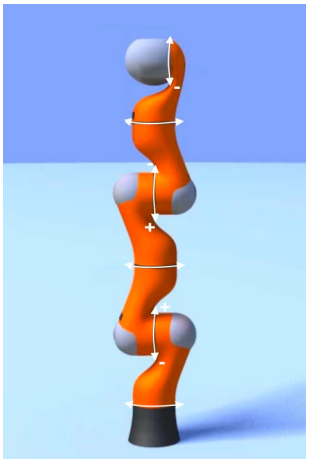


Figure 9.1: Conceptual view of the Kuka LWR robot arm, for which the method was specifically implemented. This robot arm is made of 5 articulated links of almost identical shape and appearance.

initial developments easier, not having to deal with the practical problems of noisy and mislabeled training images. The ultimate goal is however to use real images of the robot for training. In addition to the work presented in the following paper, which was entirely realized by the author of this thesis, another student, Dadhichi Shukla, worked in parallel on the same topic for his Master thesis [60]. He directly investigated the use of real training images, which proved indeed more challenging. The difficulties were due to a number of practical reasons, including noise in the images. This made the extraction of edges and of the silhouette of the links unstable across viewpoints. Another reason was the difficulty of acquiring images of the links under a good range of precisely chosen viewpoints.

The recognition of the robot arm in an image may appear as a seemingly simple task, but it showed many practical challenges. We obtained good results with the model trained on synthetic images, resulting in the capability to automatically segment images of a scene into two categories: the parts belonging to the robot, and the elements of the scene (Fig. 9.3). More developments and experiments are however needed. Ultimately, we expect the approach to be of use in the following tasks.

- Determining the explict configuration of the robot arm, i.e. the exact joint angles.
- Performing automatic hand-eye calibration, and recalibration on-the-fly (which is a tedious practical necessity).
- Using the visually-determined position of the robot arm in the manner of visual servoing, for obstacle avoidance and control directly.

As mentioned above, we also expect the recognition method to be trainable with real images of the robot instead of synthetic ones. Ideally, these images could be acquired autonomously by the robot, observing itself while exploring its range of possible movements, and thereby learning its own appearance.

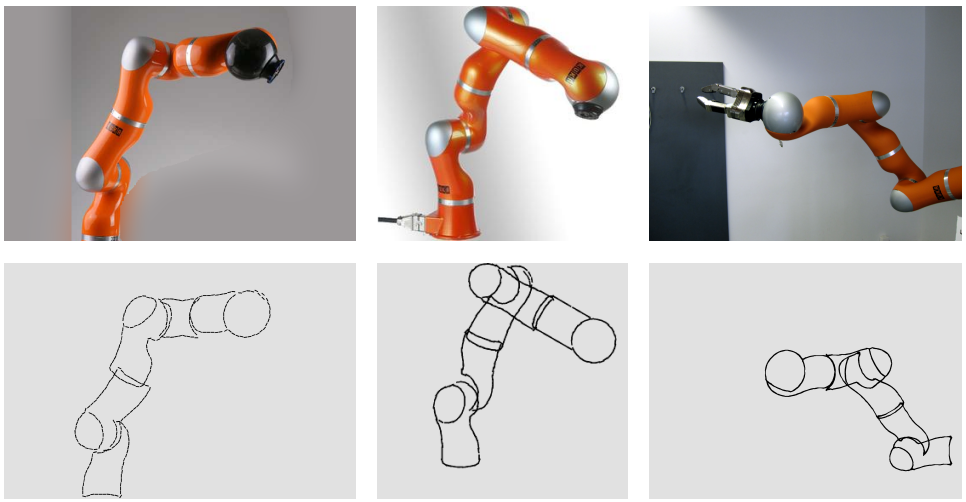


Figure 9.2: Results of pose estimation of the complete arm in three simple images (top row). After identifying a globally consistent configuration for the whole arm, we render, for visualization purposes, each link in the detected pose (bottom row).

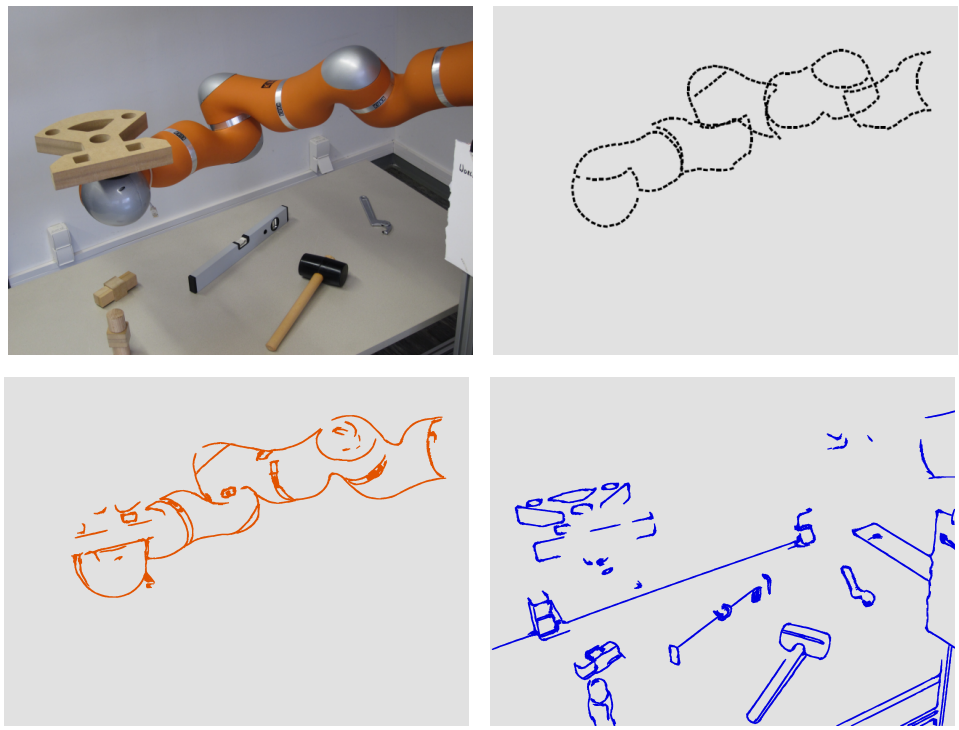


Figure 9.3: Results on a complete scene. From a single image of the scene (top left), we identify the pose of the whole arm (rendered at top right), then use this information to segment the image features (edges) into the parts belonging to the robot (bottom left) and the elements of the scene or background (bottom right).

The abstract included on the following page was presented at the *Workshop on Interactive Perception* of the 2013 *International Conference on Robotics and Automation (ICRA)*.

Markerless Self-Recognition and Segmentation of Robotic Manipulator in Still Images

ICRA 2013 Mobile Manipulation Workshop on Interactive Perception

Damien Teney
University of Liège, Belgium

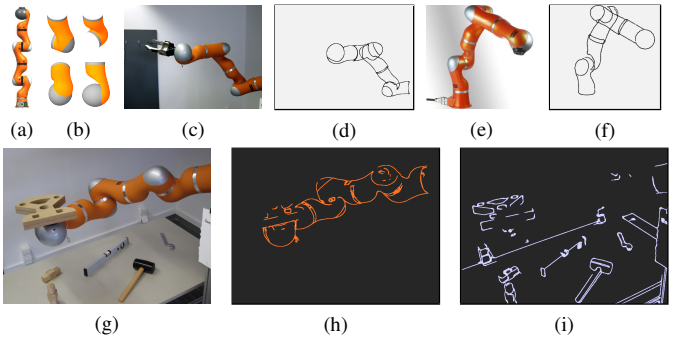
Dadhichi Shukla
University of Burgundy, France

Justus Piater
University of Innsbruck, Austria

Vision is a crucial capability for enabling robots to perceive and interact with their environment, e.g. manipulating or grasping objects. A current trend is bringing closer the aspects of interaction and perception, on the one hand by integrating visual information directly in the control process, and on the other hand, using interaction itself to help perception, allowing robots to explore their environment. In the context of manipulation, physical parts of the robot are then likely to appear in the observations, and an important capability emerges as the recognition of those parts, in order to separate the observations of the scene from those of the robot itself. Identifying the robot's own body parts in input images has been used before in different ways, helping obstacle avoidance or control directly (through visual servoing [1]). However, this is usually performed via indirect methods, tracking fiducial markers purposely attached to the robot [2], which imposes undesirable (e.g. visibility) constraints. Some recent work adresses the pose estimation of a robot manipulator *directly* [3], [4], but these methods focus on tracking the manipulator between consecutive frames, whereas the initial recognition is considered as the harder part. We propose a method for markerless, monocular recognition and pose estimation of an articulated robot arm, dealing with single images without initialization, allowing its use with unknown hand-eye calibration, imprecise kinematics or missing position feedback.

We use an existing appearance-based recognition method [5], that relies on object edges and contours, allowing the recognition of objects with few characteristic visual features (i.e. non-textured). The system is trained separately for each articulated link of the robot arm, using synthetic images of that link in known poses. The recognition of those elements proves extremely challenging, as their appearance may not offer very distinguishable visual clues, and because of the typical unstructured environment (background clutter) and possible (self-)occlusions. The initial recognition produces a set of candidate poses for each link, which are then combined, enforcing the known kinematic constraints between the links. These constraints are modeled as pairwise compatibility functions in a classical Markov random field, with a node for each link. Inference is carried out with an algorithm inspired by non-parametric belief propagation. We efficiently limit the evaluation of densities in the pose space to the discrete points proposed by the initial recognition step, thereby ensuring adequate efficiency. The algorithm ultimately recovers the pose of all the elements (links) of the arm. We can then classify the image features of the input scene as belonging to the scene or to the robot manipulator itself, simply by measuring their similarity with the training templates of the links in the identified poses. The poses of the links can also be used to recover the angles at each joint, together with the cartesian position of each element of the robot relative to the camera.

The system was implemented and tested with a Kuka Lightweight Robot arm. We considered the five internal links, of which four are completely identical in appearance, which constitutes an additional challenge. The four joints (revolute, both axial and hinge-like) are specified by the alignment of the joint axes of the adjacent links. Training images of the two types of links were generated with CAD software, at viewpoints about 30° apart. Although the recognition of individual links cannot be relied upon for any practical purposes, the pose ultimately recovered for the whole arm by the proposed algorithm was correct during most of our tests. The probabilistic inference can handle missing detections of links to some degree, as can happen with (self-)occlusions. The classification of the image features of the test image as robot/non-robot parts, as mentioned above, proved effective, and superior to using intermediate segmentation masks. Indeed, our procedure can handle occlusions, for example when the robot is manipulating an object. The capabilities offered by the whole system should help and make more robust the subsequent processing of visual data in the context of joint perception/manipulation scenarios. Future work will aim at relaxing the assumptions of known link and joint geometries, e.g. reusing existing work on autonomous learning of articulated models [6].



(a) (b) (c) (d) (e) (f)

(g) (h) (i)

(a) The Kuka LWR used in our experiments (b) Synthetic training images of individual links (c, e) Test images and (d, f) rendering of the training templates as recognized for each link (g-i) After recognition of the robot arm, image features are classified as "robot" (orange) and "non-robot" (blue).

Research supported through European Commission's FP7 projects Xperience and IntellAct, and by the Belgian National Fund for Scientific Research.

[1] M. Prats, P. Martinet, A. del Pobil, and S. Lee, "Robotic execution of everyday tasks by means of external vision/force control," *Intelligent Service Robotics*, vol. 1, no. 3, pp. 253–266, 2008.

[2] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann, "Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot," in *ICRA*, 2010.

[3] J. J. Sorribes, M. Prats, and A. Morales, "Visual tracking of a jaw gripper based on articulated 3D models for grasping," in *ICRA*, 2010.

[4] X. Gratal, J. Romero, and D. Kragic, "Virtual visual servoing for real-time robot pose estimation," in *IFAC*, 2011.

[5] D. Teney and J. Piater, "Continuous pose estimation in 2D images at instance and category levels," in *Comp. and Rob. Vis.*, 2013.

[6] D. Katz and O. Brock, "Extracting planar kinematic models using interactive perception," in *Unifying Perspectives In Computational and Robot Vision*, no. 8. Springer Verlag, May 2008, pp. 11–23.

This article is part of Chapter 9: *Application to the recognition of a robotic arm* (p.127)
It originally appeared in: *Workshop on Interactive Perception* at the *2013 International Conference on Robotics and Automation*

Chapter 10

Conclusions and perspectives

10.1 Summary of contributions

This thesis presented a number of contributions to the field of computer vision, in the form of new approaches and methods related to the recognition of objects in 2D images. These contributions are summarized in the following points.

- The application of probabilistic 3D object models [41] to the problem of pose estimation in 2D images, with an appropriate algorithm to identify the pose in which the reprojected object maximizes its match with the test image (Chapter 2).
- A method for multiview 3D reconstruction from calibrated images, based on an original formulation of the reconstruction as a probability distribution in the reconstructed 3D space, which leads to a reconstruction method based on sampling (Chapter 3).
- In addition to the reconstruction of individual points, an algorithm to reconstruct continuous curves, by identifying ridges of local maxima of the probability density in the 3D space (Chapter 3).
- An approach to model the 2D appearance of objects as probability distributions of image features, similarly applicable to both specific objects and object categories (Chapter 5).
- A suitable measure of similarity of such appearance models with a test image (Chapter 5).
- An efficient voting-based algorithm to identify the local maxima of this measure of similarity, in order to perform detection in cluttered images (Chapter 8).
- A sampling method for these distributions of features particularly suitable to models learned from noisy and cluttered training examples, which focuses on the main modes of the distribution (Chapter 8).
- The application of these distributions of features to "edge points" (Chapter 8).
- The application of these distributions of features to "gradient points", with techniques specifically designed to provide some invariance to lighting (Chapter 7).

- A procedure to identify (with optical flow) and store, within our multiview model, the changes of appearance between discrete, trained viewpoints, which leads to a generative model capable of interpolating the appearance of the object of interest at arbitrary (possibly unseen) viewpoints (Chapter 6).
- A first method to perform continuous pose estimation, which works by fitting, in the pose space, a distribution over measures of similarity between the test image and different (trained) viewpoints (Chapter 5).
- A second method to perform continuous pose estimation, which uses the generative model mentionned above to perform an explicit optimization of the pose, maximizing the similarity between the generated view and the test image (Chapter 6).
- A procedure for learning weights assigned to the training data of our multiview model (from the training data itself or from negative examples, if available), leading to significant improvements for the tasks of both detection and pose estimation (Chapter 8).
- An application of the overall method for object recognition to articulated objects, implemented specifically for a robot arm; this leads to the capability of recognizing the robot arm in 2D images, e.g. for monitoring a manipulation scene. The method can then segment such images into parts of the robot and elements of the scene (Chapter 9).

Our simple approach to modeling object appearance showed remarkable performance on a number of different tasks with benchmark datasets. We acknowledged some of its limitations, such as not modeling co-occurrences of image features. This limits the applicability of our approach to object categories with clear visual characteristics, and not categories defined semantically for example. We also do not claim to compete with more flexible models such as part-based models [31–33]. One possible avenue for future research is to combine, or integrate some of our concepts within such of those existing methods.

10.2 Perspectives

An important aspect of the work presented in this thesis, in our view, is to consider different tasks together, e.g. pose estimation and shape matching, and to propose common resolution methods. The more common approach is to study these tasks as separate research topics, and to evaluate methods in the conditions imposed by the available "benchmark" datasets. This usual approach may be suboptimal or counterproductive to the general advances in the field of computer vision. Even though, there cannot be a single best method suitable to the variety of challenges still to solve in this field, and the procedures proposed within this thesis are merely alternative ideas on how to address some classical problems with simple approaches. The future grand advances in object recognition are likely to be brought by radically different methods. For example, the recognition of objects in very complex scenes, such as those represented in

the Pascal Visual Object Classes Challenge [61], requires much more than just modeling the appearance of object categories or recognizing particular shapes in images. The recognition of such complex objects must involve the overall visual context and require, ideally, a complete understanding of the scene as a prerequisite to recognizing its elements. This fact is acknowledged by the most successful methods applied to these datasets (e.g. [31, 62, 63]), which indeed use context, and leverage the advances in the field of machine learning. Other methods have been proposed recently to address the recovery of the overall geometry of complete scenes, and will also help moving towards a similar direction.

On the topic of the autonomous learning of visual object models, as needed by robots that must evolve and adapt to human, unstructured environments, we also believe in dramatic improvements from the use of currently underexploited sources of information. Nonvisual information, or *metadata*, as coined by previous authors (e.g. in [14]) could be retrieved from the internet and inform the recognition method about various properties of the objects. This information could be related to the potential use of the objects (as to help in a purpose-oriented search for objects), or to the environment in which the objects are encountered (and likely to be encountered). This largely crosses the field of robotics, and will help building object models relevant not only to their visual recognition, but also to their possible uses, or *affordances*, as called in the domain of robotics.

A last example of an extraneous source of information for visual recognition is active vision and interactive perception. Within this paradigm, the observer (the robot) is free to observe the scene from chosen viewpoints and to interact with it in order to resolve ambiguities. This can provide powerful additional clues compared to simple static photographs. More and more work has emerged in that area recently, e.g. to help building models of articulated objects [36], or to help segmenting scenes into their different elements [37, 40].

The research directions proposed above go well beyond the scope of this thesis, but are likely to lead to significant improvements for the tasks that we considered during our work. These research directions thus represent promising avenues for future developments, which will hopefully help the research to move a few steps forwards in the field of Computer Vision.

References of publications included in this thesis

[1] D. Teney, J. Piater, Probabilistic Object Models for Pose Estimation in 2D Images, in: DAGM, vol. 6835/2011 of *LNCS*, Springer, Heidelberg, 336–345, 2011 [Chapter 2].

[2] D. Teney, J. Piater, Sampling-based Multiview Reconstruction without Correspondences for 3D Edges, in: 3DimPVT, IEEE, 160–167, 2012 [Chapter 3].

[3] D. Teney, J. Piater, Generalized Exemplar-Based Full Pose Estimation from 2D Images without Correspondences, in: Digital Image Computing: Techniques and Applications, 2012 [Chapter 4].

[4] D. Teney, J. Piater, Continuous Pose Estimation in 2D Images at Instance and Category Levels, in: Computer and Robot Vision, 2013 [Chapter 5].

[5] D. Teney, J. Piater, Modeling Pose/Appearance Relations for Improved Object Localization and Pose Estimation in 2D images, in: 6th Iberian Conference on Pattern Recognition and Image Analysis, vol. 7887 of *LNCS*, Springer, 59–68, 2013 [Chapter 6].

[6] D. Teney, J. Piater, Probabilistic Templates of Dense Image Features for Object Detection and Recognition, to be submitted, 2013 [Chapter 7].

[7] D. Teney, J. Piater, Multi-view Feature Distributions for Object Detection and Continuous Pose Estimation, in: Computer Vision and Image Understanding, submitted, 2013 [Chapter 8].

[8] D. Teney, D. Shukla, J. Piater, Markerless Self-Recognition and Segmentation of Robotic Manipulator in Still Images, in: Mobile Manipulation Workshop on Interactive Perception, workshop at ICRA, 2013 [Chapter 9].

Bibliography

[9] V. Ferrari, F. Jurie, C. Schmid, From Images to Shape Models for Object Detection, International Journal of Computer Vision (IJCV) 87 (3) (2010) 284–303.

[10] S. Maji, J. Malik, Object Detection using a Max-margin Hough Transform, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[11] S. Ravishankar, A. Jain, A. Mittal, Multi-stage Contour Based Detection of Deformable Objects, in: IEEE European Conference on Computer Vision (ECCV), 483–496, 2008.

[12] S. Savarese, L. Fei-Fei, 3D Generic Object Categorization, Localization and Pose Estimation, in: IEEE International Conference on Computer Vision (ICCV), ISSN 1550-5499, 2007.

[13] F. Rothganger, S. Lazebnik, C. Schmid, J. Ponce, 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints, International Journal of Computer Vision (IJCV) 66 (3) (2006) 231–259.

[14] A. Collet Romea, Lifelong Robotic Object Perception, Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2012.

[15] C. Harris, M. Stephens, A Combined Corner and Edge Detector, in: In Proc. of Fourth Alvey Vision Conference, 147–151, 1988.

[16] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of Interest Point Detectors, International Journal of Computer Vision (IJCV) 37 (2) (2000) 151–172, ISSN 0920-5691.

[17] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust Wide-baseline Stereo from Maximally Stable Extremal Regions, Image and Vision Computing 22 (10) (2004) 761 – 767, ISSN 0262-8856.

[18] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 8 (1986) 679–698, ISSN 0162-8828.

[19] D. R. Martin, C. C. Fowlkes, J. Malik, Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 26 (5) (2004) 530–549, ISSN 0162-8828.

[20] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision (IJCV) 60 (2) (2004) 91–110.

[21] N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 886–893, 2005.

[22] A. C. Berg, J. Malik, Geometric Blur for Template Matching, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, 607–614, 2001.

[23] C. Gu, J. J. Lim, P. Arbelaez, J. Malik, Recognition using Regions, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[24] J. Mutch, D. Lowe, Multiclass Object Recognition with Sparse, Localized Features, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, ISSN 1063-6919, 11–18, 2006.

[25] C. Huang, H. Ai, Y. Li, S. Lao, Vector Boosting for Rotation Invariant Multi-view Face Detection, in: IEEE International Conference on Computer Vision (ICCV), vol. 1, ISSN 1550-5499, 446–453 Vol. 1, 2005.

[26] A. Torralba, K. Murphy, W. Freeman, Sharing Visual Features for Multiclass and Multiview Object Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29 (5) (2007) 854–869, ISSN 0162-8828.

[27] D.H., Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, Pattern Recognition 13 (2) (1981) 111 – 122, ISSN 0031-3203.

[28] M. Pierre, P. Pietro, A Probabilistic Cascade of Detectors for Individual Object Recognition, in: IEEE European Conference on Computer Vision (ECCV), 2008.

[29] V. Ferrari, T. Tuytelaars, L. Van Gool, Object Detection by Contour Segment Networks, in: IEEE European Conference on Computer Vision (ECCV), ISBN 3-540-33836-5, 978-3-540-33836-9, 14–28, 2006.

[30] V. Ferrari, F. Jurie, C. Schmid, Accurate Object Detection with Deformable Shape Models Learnt from Images, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 1–8, 2007.

[31] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part-Based Models, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (9) (2010) 1627–1645, ISSN 0162-8828.

[32] S. Fidler, M. Boben, A. Leonardis, Learning Hierarchical Compositional Representations of Object Structure, in: Object Categorization: Computer and Human Vision Perspectives, Cambridge University Press, 2009.

[33] B. Leibe, A. Leonardis, B. Schiele, Robust Object Detection with Interleaved Categorization and Segmentation, International Journal of Computer Vision 77 (1-3) (2008) 259–289, ISSN 0920-5691.

[34] H. Harzallah, F. Jurie, C. Schmid, Combining Efficient Object Localization and Image Classification, in: Computer Vision, 2009 IEEE 12th International Conference on, ISSN 1550-5499, 237–244, 2009.

[35] P. Viola, M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, ISSN 1063-6919, I–511–I–518 vol.1, 2001.

[36] D. Katz, O. Brock, Extracting Planar Kinematic Models Using Interactive Perception (8) (2008) 11–23, ISSN 1876-1100.

[37] D. Katz, M. Kazemi, J. A. D. Bagnell, A. T. Stentz, Interactive Segmentation, Tracking, and Kinematic Modeling of Unknown Articulated Objects, Tech. Rep. CMU-RI-TR-12-06, Robotics Institute, Pittsburgh, PA, 2012.

[38] N. Govender, J. Claassens, F. Nicolls, J. Warrell, Active Object Recognition using Vocabulary Trees, in: IEEE Workshop on Robot Vision (WORV), 20–26, 2013.

[39] C. Laporte, T. Arbel, Efficient Discriminant Viewpoint Selection for Active Bayesian Recognition, International Journal of Computer Vision (IJCV) 68 (3) (2006) 267–287, ISSN 0920-5691.

[40] A. N. L. N. J. K. T. P. G. Sankaran, B., K. Daniilidis, Hypothesis Testing Framework for Active Object Detection, in: IEEE International Conference on Robotics and Automation, 2013.

[41] R. Detry, N. Pugeault, J. Piater, A Probabilistic Framework for 3D Visual Object Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 31 (10) (2009) 1790–1803.

[42] N. Krüger, M. Lappe, F. Wörgötter, Biologically Motivated Multi-modal Processing of Visual Primitives, The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour 1 (5) (2004) 417–428.

[43] N. Pugeault, Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation, Vdm Verlag Dr. Müller, 2008.

[44] A. Dempster, N. Laird, D. Rubin, et al., Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statistical Society. Series B (Methodological) 39 (1) (1977) 1–38.

[45] E. Alpaydin, Introduction to Machine Learning, The MIT Press, 2004.

[46] B. W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall/CRC, 1986.

[47] K. Mardia, P. Jupp, Directional Statistics, Wiley Series in Probability and Statistics, Wiley, ISBN 9780470317815, 2009.

[48] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, ISBN 0-89871-295-5, 1992.

[49] R. E. Caflisch, Monte Carlo and Quasi-Monte Carlo Methods, Acta Numerica 7 (1998) 1–49, ISSN 1474-0508.

[50] R. Detry, J. Piater, Continuous Surface-point Distributions for 3D Object Pose Estimation and Recognition, in: Asian Conference on Computer Vision (ACCV), 2010.

[51] S. Edelmann, H. Bülthoff, Modeling Human Visual Object Recognition, in: International Joint Conference on Neural Networks (IJCNN), IEEE, ISBN 0-7803-0559-0, 37–42, 1992.

[52] A. Yershova, S. Jain, S. M. LaValle, J. C. Mitchell, Generating Uniform Incremental Grids on $SO(3)$ Using the Hopf Fibration, International Journal of Robotic Research 29 (7) (2010) 801–812.

[53] A. S. Szalay, J. Gray, G. Fekete, P. Z. Kunszt, P. Kukol, A. Thakar, Indexing the Sphere with the Hierarchical Triangular Mesh, Tech. Rep., Microsoft Research, 2005.

[54] M. Ozuysal, V. Lepetit, P. Fua, Pose estimation for Category Specific Multiview Object Localization, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 2009.

[55] A. Frome, Y. Singer, F. Sha, J. Malik, Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification, in: IEEE International Conference on Computer Vision (ICCV), 2007.

[56] P. Yarlagadda, B. Ommer, From Meaningful Contours to Discriminative Object Shape, in: IEEE European Conference on Computer Vision (ECCV), 2012.

[57] M. Sun, H. Su, S. Savarese, L. Fei-Fei, A Multi-View Probabilistic Model for 3D Object Classes, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 2009.

[58] S. Savarese, L. Fei-Fei, View Synthesis for Recognizing unseen Poses of Object Classes, in: IEEE European Conference on Computer Vision (ECCV), 2008.

[59] K. Welke, J. Issac, D. Schiebener, T. Asfour, R. Dillmann, Autonomous Acquisition of Visual Multi-View Object Representations for Object Recognition on a Humanoid Robot, in: ICRA, 2010.

[60] D. Shukla, Learning a Visual Model of a Robot Arm, Master's thesis, University of Burgundy, France and University of Innsbruck, Austria, 2013.

[61] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The Pascal Visual Object Classes (VOC) Challenge, International Journal of Computer Vision (IJCV) 88 (2) (2010) 303–338.

[62] S. Maji, L. Bourdev, J. Malik, Action Recognition from a Distributed Representation of Pose and Appearance, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 3177–3184, 2011.

[63] Z. Song, Q. Chen, Z. Huang, Y. Hua, S. Yan, Contextualizing Object Detection and Classification, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN 1063-6919, 1585–1592, 2011.